

## A Generalized Reliability Mechanism with minimum -Replica by PRCR

<sup>1</sup>M. Sravan Kumar, <sup>2</sup> K. Ramesh, <sup>3</sup> T. Subramanyam

<sup>1</sup> PG Scholar, <sup>2</sup> Associate Professor, <sup>3</sup> Assistant Professor

<sup>1,2,3</sup> PDIT, JNTUA University.

<sup>1</sup> [sravan.manubolu@gmail.com](mailto:sravan.manubolu@gmail.com)

<sup>2</sup> [kumbha.ramesh@gmail.com](mailto:kumbha.ramesh@gmail.com)

<sup>3</sup> [subbu637@gmail.com](mailto:subbu637@gmail.com)

### Abstract

The rapid progress of technology and large scale use of internet has resulted in generation of an enormous amount of data. In the recent years, cloud computing has become popular among various fields like information technology and various business enterprises. The fundamental use of the cloud now a days is for storing information and sharing the resources. Cloud is another way to store expensive measure of information. Cloud provides the storage space and offering to use this information to various clients. Additionally it is technique for pay according to we utilize. The two principle concerns in current cloud storage systems are providing the reliable data and storage cost. To ensure data reliability, current cloud systems uses multi-replica strategy (typically three replicas), which incurs a huge storage space, on the effect it leads to more storage cost for data-intensive applications in the cloud. To minimize the storage space, in this paper we proposed a cost effective data reliability mechanism called Proactive Replica Checking for Reliability (PRCR) by using a generalized data reliability model. PRCR guarantees the reliability with the reduced replication factor, which can likewise reduce the storage cost for replication based methodologies. Contrasting to the traditional three – replica strategy, PRCR can diminish the storage space utilization by one-third to two-thirds of the current storage space, thus fundamentally bringing down the storage cost.

**Keywords:** cloud computing, minimum data replication, PRCR, Cost-effective storage, variable disk failure rates.

### 1. INTRODUCTION

The rapid progress of technology and large scale use of internet has resulted in generation of an enormous amount of data. With the emergence of internet as a highly reliable system, the organizations have realized the importance of outsourcing of the activities like, computation, processing etc. as a utility. Large Organizations have begun to outsource services like platforms, network, and storage as it is perceived as a cost effective option rather than investing heavily on these requirements within. The cloud could now provide features like broad network access, resource pooling and rapid elasticity [13]. However there are certain disadvantages pertaining to the safety, security and privacy of the data stored.

#### 1.1 Emergence of Storage in Cloud

With the rise of social media, high use of data acquisition devices in scientific research, news, and web based applications; there is a huge amount of generation of structured and unstructured data. The availability of cloud storage services is becoming a popular option for consumers to store data that is accessible via a range of devices such as personal computers, tablets and mobile phones [14]. Smaller organizations having large web applications have started to move their data to the cloud storage system like Drop Box, Giga spaces, and Elephant Drive. Cloud storage system consolidates large numbers of geographically distributed

computers into a single storage pool and provides large capacity high performance storage service at low cost [15]. Distributed File Systems is a critical component of Cloud Storage Systems. Google File Systems (GFS), Hadoop Distributed File System (HDFS) , & Amazon Simple Storage Service (S3) are the working examples of distributed file systems. These file systems are typically used for efficient and reliable storage of large data sets.

As the data storage is accomplished by employing large number of data nodes the probability of failure of these nodes is quite high. This issue is augmented by unreliable connectivity and limited bandwidth of the network. Therefore improving the data availability and reliability has become a prominent challenge in cloud storage.

#### 1.2 The Need for Replication

Data replication has been widely used, as a mean to increase data availability and reliability of cloud storage systems. Storage Systems like HDFS, GFS, Amazon S3, employ conventional three replication strategy where three copies of the data including original copy are stored on geographically distributed servers or nodes [5].

Data replication algorithm can be divided into two broad categories, Static Replication and Dynamic Replication [6]. In static replication strategy, number of replicas and their location is predetermined and fixed; on the other hand

dynamic replication automatically creates replicas according to changing access patterns, and location of replicas can be changed dynamically based on capacity of nodes [3].

Dynamic replication strategies have three important issues that must be addressed. They are as below:

1. Selection of the data file to be replicated and the decision regarding when to replicate so as to meet users requirement such as waiting time reduction and data access speeding up. Proper selection of data files if not done then, it will lead unnecessary consumption of storage space, thereby increasing the storage cost. Similarly too early replication of a data file will lead to same issues.

2. Finding the optimal number of new replicas, to be created in the system for meeting the system availability, reliability requirement and the cost of replica maintenance, because after a certain point of time increasing the number of replicas does not increase the availability but might bring unnecessary spending like extra storage space consumption and associated storage cost [15].

3. Placement of replicas in a balanced way to meet system task successful execution rate and bandwidth consumption requirement. Idea is to place the replicas closer to the user, in order to minimize the response time, and thus job execution time. This will increase the throughput of the system

The data which are stored in the cloud ought to initially meet the reliability criteria with high efficiency and cost effectiveness. High reliability is essential for the typical cloud systems. Be that as it may, accomplish the reliability with a sensible cost can be challenging one, particularly in large scale systems.

The size of the cloud storages is growing at a dramatic speed. Under the analysis, the data storage in the cloud almost reached 1 ZB, while more data are stored or processed in their journey. Interim, with the development of the cloud computing paradigm, cloud-based applications have advanced their demand the cloud storage [1]. While the necessity of the data reliability should be met in the first place and the data in the cloud should be put away with higher cost-viability.

Data reliability is the main concern in the cloud and it is defined as the likelihood of one data item available in the cloud system for a specific period. It is a critical issue in the cloud storage systems, which shows the capacity of keeping the data consistently and accessible fully by the client/user. Because of the quick growth in the cloud data, providing the data reliably has turned in to a challenging task. At present, data replication is a standard method for providing the data reliability. Nonetheless, the consumption of the storage space with the help of replication methodology brings about the immense cost, and it is trusted that cost would be passed on to the clients in the end as a result.

In this paper, we present a new cost-effective data reliability management mechanism based on the proactive replica checking called PRCR for decreasing the storage space consumption, hence subsequently reducing the storage cost

for data-intensive applications in the cloud systems. The main goal of PRCR algorithm is to decrease the number of replicas stored in the cloud while meeting the data reliability requirement. Contrasted with the traditional data reliability mechanisms in the cloud application, the PRCR algorithm contains the following features:

1. As the First level, the PRCR algorithm is guaranteed to provide the data reliability.
2. PRCR is capable of providing data reliability management in a cost-effective manner. By applying PRCR, data reliability is guaranteed with at most two replicas stored in the cloud.

By applying benchmarks from Amazon web services, we assess the efficiency of PRCR nodes and simulate the reliability management process utilizing PRCR with the data created by data-intensive storage applications. The outcomes are contrasted with the traditional three-replica mechanism. It is observed that the PRCR mechanism can reduce the storage space from 33% to 66% of the storage space than the existing methods used in the cloud, subsequently, it also decrease the storage cost also.

The rest of the paper is organized as follows. The related works on data reliability, data replication and cost-effective data storage are addressed in Section 2. The proposed work in which the concepts of PRCR and its working procedure are explained in section 3. The Storage duration prediction algorithm and Meta data distribution algorithms are discussed in section 4. Results and Discussions are discussed in Section 5. Finally, conclusions and future work are summarized in Section 6.

## 2. LITERATURE SURVEY

S. Ramabhadran et.al. [6]. have studied and assumed that the failure of each disk is constant in the exponential data reliability model. For example, the existing studies analyze that the Markov chain model assumes that the failure rate of each disk in the cloud or any other storage system is the same. In reality, the constant disk failure rate cannot explain the overall phenomena. It has been observed the mostly the failure rate of the hard disk drives follows a "bath tub" curve model in the failure rate is higher in the disk early life, and drops during some period, and remains constant for the remainder of the disk life span and increase again at the end of the disk's lifetime.

The disk failure probability does not follow an exponential distribution. By fixing this inconsistency IDEMA viz. International Disk Drive Equipment and Materials Association proposed a satisfactory presentation for disk failure rates that, the life span of each disk is divided into various life stages with various failure rates. [10]. In a nine-month investigation, Google also gets the results inconsistent with this model. This paper concentrates the disk failure rates with IDEMA style [11], in which the lifespan of each disk is varied accordingly. Many efforts and research are performed for providing the data reliability as within the case of software aspect also.

A.Gharaibeh, B.Balasubramanian[4] studied and analyzed that Data replication is considered as the dominant approach in the cloud storage system for providing the data reliability and recent works on the large-scale cloud storage systems followed the multi-replication strategy to ensure the data reliability. In the field of cloud computing, data replication has been a widely adopted technique in the commercial distributed storage systems. Amazon S3, Google File System (GFS) [11], HDFS are the typical examples that are following three replica strategy.

Although the data replication strategy is used by many commercial cloud storage service providers, Amazon S3 proposed Reduced Redundancy Storage (RRS) to address the data redundancy issue to decrease the storage consumption [10]. The problem with RRS is it sacrifices the data reliability. With the help of low level of reliability only provided. An erasure coded storage system also presented and implemented by K.V Rashmi [3]. In the approach, the data is divided into various blocks and store them with additional erasure coding blocks. By using this kind of technique reliability is assured but there is a computation overhead in encoding and decoding the data. Hence, In the case of data intensive applications, erasure coding methods are not the best solution compared with the replication-based mechanisms.

### 3. PROPOSED WORK

The PRCR runs on the virtual machines (VM) in the cloud environment. This VM is responsible for running the user interface, PRCR node and to conduct the proactive replication checking separately. The main responsibility of the user interface is to determine the minimum replication factor. It additionally makes the duplicate (replicas) if necessary and distributes the metadata of records. At whatever point the first replica of the file is created or uploaded in the cloud stage it finds the minimum number of replica depending on the storage duration that is either short period or long period of existence of the file in the cloud. With this principle, it makes the replication either one or two. If one replica is not sufficient for one particular file to store the data then the user interface calls the cloud to create the second replica to the corresponding file. When the second replication of that file is created, it also creates one metadata file and this will be distributed to the suitable PRCR node. The metadata has added up to six sorts of attributes such as file Identifier, time stamp, data reliability required, expected storage duration, checking interval and the address of the replica. The record of file ID and the replica address is automatically given when the first or second copy of the files are made.

#### 3.1 Proactive Replica Checking for Reliability (PRCR)

The principle thought to use proactive replica checking is to propose a cost-effective data reliability management mechanism for cloud data storage. The PRCR utilize the well-known property of exponential distribution called the memoryless property. In PRCR the information in the cloud is organized into various types according to its expected storage time and reliability management. For those data

items, which are required for short term storage are sufficient to maintain a single copy to provide the data reliability. For those data items, which are required for long term or regularly used, maintains three replica strategy for providing the reliability. Consider an example, there are 500 files each of 1 Mb size. Typically, with the conventional strategy we should need to maintain three replicas for each type, it requires totally 1500 MB storage space (500 \* 3 copies of each), results from a huge storage cost. As with two kinds of data types, assume there are 300 files among 500 are critical and 200 files are for the short term use. If we consider this scenario  $300 * 3 + 200 * 1 = 1100$  MB of storage space required, which will save 400 MB compared with the conventional three-replica strategy. The proactive replica checking is done at regular intervals to check the presence of the replicas. If in some cases both the replicas may be lost, the probability of this situation is incorporated in the data reliability model in its earliest. PRCR ensures that the data loss rate is maximum of 0.01 percent of the total data per year

PRCR keeps running on the top virtual layer of the cloud and overseas data which are stored as the file in various data centers. Figure 1. Demonstrates the architecture of the Proactive Replica Checking for Reliability (PRCR). There are two noteworthy parts in the architecture are PRCR node and UI (User Interface)

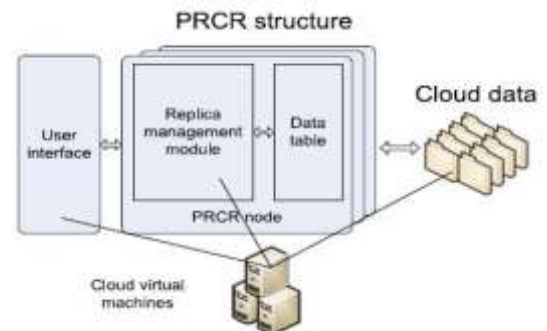


Figure 1: PRCR Architecture

**PRCR node:** The main segment in the PRCR architectures is the PRCR node. The main responsibility of this node is the management of the replicas. As will be specified later, as per the number of files in the cloud, PRCR may contain at least one or more PRCR nodes and each one is independent together. The PRCR node is made of two noteworthy components inside. One is data table and the second is replica management module.

**Data Table:** It keeps track the metadata of all data that each PRCR node manages. For each file in the data storage, four metadata attributes are maintained in the data table. They are File Identifier (file ID), scan interval, the timestamp value (time stamp) address of the replica (replica address). File ID is used to uniquely identify the file. The scan interval is the time between the two replica checking of each file. Timestamp records the time whenever the replica checking is done on one file. PRCR can decide the files that need to be checked depending on the scan interval and timestamp and based on the replica address all the replicas are found. Here each round of the scan is considered as a scan cycle. In each scan cycle, metadata is sequentially examined once.

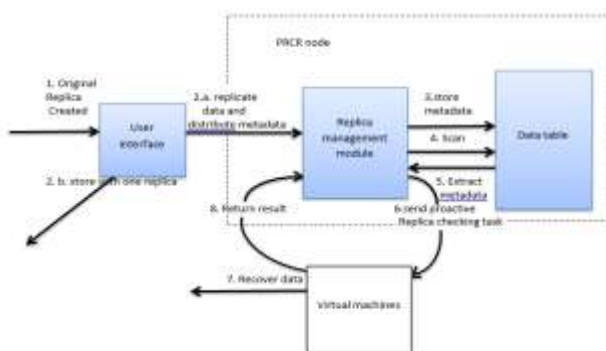


**Replica Management Module:** It is the control component of the PRCR node, which is responsible for managing the metadata in the data table and cooperating with the Cloud computing instance2 to process the checking tasks. In each scan cycle, the replica management module scans the metadata in the data table and determines whether the file needs to be checked. For files that need to be checked, the replica management module extracts their metadata from the data table and sends these metadata to the Cloud computing instances. After the Cloud computing instances have finished the checking tasks, the replica management module receives the returned results and conducts further actions accordingly. In particular, if any replica is not available, the replica management module sends a request for replication to the Cloud which creates a new replica of the data and updates the data table accordingly.

**User interface:** It is a very important component of PRCR, which is responsible for justifying reliability management requests and distributing accepted requests to different PRCR nodes. A reliability management request is a request for PRCR to manage the file. In the request, the metadata of the file is required. Despite the four types of metadata attributes in the data table, a metadata attribute which is called the expected storage duration is needed. It is an optional metadata attribute of the file, which indicates the storage duration that the user expects. According to the expected storage duration, the user interface is able to justify whether the file needs to be managed by PRCR or not. In particular, if such management is unnecessary, the reliability management request is declined and the file is stored with only one replica in the Cloud.

### 3.2 Working Procedure of PRCR

We illustrate the working process of PRCR in the Cloud by following the life cycle of a file and the same working process shown in Figure 2.



**Figure 2. Working Process of PRCR**

1. When the original file of the replica is created, the actual process begins at this time. Based on the issues such as disk failure rate, storage space, data reliability, the user interface will determine to store the file with one copy or two copies (replicas).

2. According to the calculation in the user interface, if one replica cannot satisfy the data reliability and storage duration requirements of the file, the user interface creates a

second replica by calling Cloud services and calculates the checking interval(s) of the file. Its metadata is then distributed to the appropriate PRCR node (2). If one replica is sufficient, only the original replica is stored and the metadata of the file is not created (9).

3. The attributes related to metadata are stored in the data table of the PRCR node.

4. Metadata is scanned periodically according to the scan cycle of the PRCR node. According to file's time stamp and the current checking interval, PRCR determines whether proactive replica checking is needed.

5. If proactive replica checking is needed, the replica management module obtains the metadata of the file from the data table.

6. The replica management module assigns the proactive replica checking task to one of the Cloud virtual machines for proactive replica checking. The Cloud virtual machine executes the task, in which both replicas of the file are checked.

7. The Cloud virtual machine conducts further action according to the result of the proactive replica checking task: if both replicas are alive or lost, go to step 8; if only one replica is lost, the virtual machine calls the Cloud services to generate a new replica based on the replica that is alive.

8. The Cloud virtual machine returns the result of the proactive replica checking task, while in the data table, the time stamp and checking interval(s) are updated. Specifically, step (1) if both replicas are not lost, the next checking interval is put forward as the current checking interval; and step(2) if a replica is lost and recovered on a new disk, the new replica address is stored and all the checking interval(s) are recalculated. Otherwise, further steps could be conducted, for example, a data loss alert could be issued.

9. Steps 4 to 8 form a continuous loop until the expected storage duration is reached or the file is deleted. If the expected storage duration is reached, either the storage user could renew the PRCR service or PRCR could delete the metadata of the file and stop the proactive replica checking process.

### 3.3 Data Reliability Model

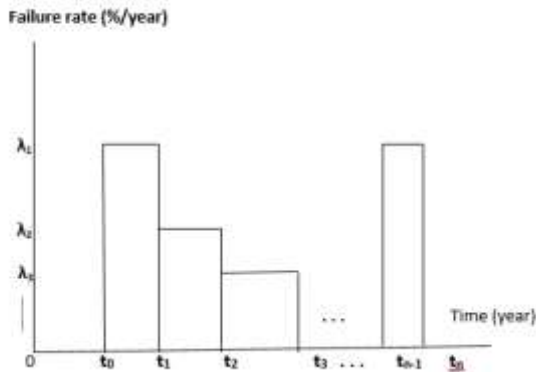
The current cloud Systems use the conventional three-replica strategy, which leads to huge storage space. There is another way to provide data reliability by reducing it into fewer replicas. A mathematical model describes the possibility of providing the reliability with the constant and variable failure rate of the disks. In the constant Disk failure rate, it follows the exponential follows the exponential distribution with a constant disk failure rate. In that case, the reliability of a disk over period T (i.e., one disk failure probability) can be expressed as R(T), where:

$$R(T) = e^{-\lambda T} \quad (1)$$

The replicas stored in the disk should have the same reliability as the disk. Therefore, (1) is also applicable for calculating the reliability of a single replica when the disk failure rate is constant. Therefore, R(T) also indicates the data reliability, i.e., the probability of the replica survives

over period T with as the disk failure rate. According to the IDEMA standard, when the disk failure rate is a variable, we assume the disk failure rate pattern contains several life stages, and in each life stage of a disk, the disk failure rate does not change. This is called variable disk failure rate and in Figure 3 it shows that the failure rate pattern of disk D between time  $t_0$  and  $t_n$ . The Reliability in the variable disk failure rate is defined as:

$$R(T) = e^{-\lambda T} \quad (2)$$



**Figure 3: The failure rate pattern of disk D between time  $t_0$  and  $t_n$**

Equation (1) and Equation (2) denotes the reliability of storing one replica in constant and variable disk failure rate. In this paper we proposed, a generalized data reliability model with a variable disk failure rate for multiple replicas is proposed and it can be defined as:

$$R(T_k) = 1 - \prod_{i=1}^k 1 - e^{-\lambda_i T} \quad (3)$$

Equation (3) reveals the relationship between data reliability, the number of replicas, disk failure rates, and storage duration. If the number of replicas and the failure rates of disks is known, the relationship between storage duration and data reliability can then be derived. It can be seen that (2) is a special case of (3) when  $k=1$ .

## 4. ALGORITHMS

### 4.1 Algorithms used in Proactive Replica Checking:

There are mainly two algorithms are used in the whole working process in PRCR. The first algorithm is storage duration prediction algorithm which is mainly used to determine the number of replicas is to be stored and to find the longest storage duration of each file. This can be shown in Figure 3. The second one is metadata storage distribution algorithm which is shown in Figure 4.

#### 4.1.1 Storage Duration Prediction Algorithm:

The Figure Shows the Pseudo code for the storage duration prediction algorithm. Where ET is the expected storage duration, P1, P2 are the failure rates of the disk 1 and disk 2 respectively. StartT is the time when the replica is being to be stored. The output SDS denotes the set of all the longest

storage durations. The algorithm first calculates the average failure rate of the file stored on disk 1 for the duration of ET (Step 1). According to this value, it determines the number of replicas that need to be stored, i.e., to store the file with one replica or two (step 2). If two replicas need to be stored, the algorithm calculates all LSDWPs throughout the expected storage duration of the file in one go and returns the LSDWPs set as a result (Steps3-11).

#### Algorithm: Data Reliability Prediction Algorithm

Input: ET// Expected Time

RR (1) // Reliability Requirement

P1, P2 // Disk Failure Rate Patterns of the Disk 1 and Disk2.

StartT // Start Time

Output: SDS

Step 1: Calculate the Average Failure Rate (P1, StartT, ET)

Step 2: if (check the number of replicas needed) {

Step 3: Assign T=StartT

Step 4: While (T<=ET+StartT) {

Step 5: Obtain the Piecewise Linear Function for P1

Step 6: Obtain the piecewise linear function for P2

Step 7: obtain the result from equation (8)

Step 8: assign SD as the Square Root of The result of (8)

Step 9: T=T+SD

Step 10: SDS = SD

Step 11 :} return SDS

Step 12 :} else return -1

#### Figure 4: Storage duration Prediction Algorithm

#### 4.1.2 Metadata Storage Distribution Algorithm:

The metadata distribution algorithm is conducted at the user interface of PRCR. Figure 5 shows the pseudo code of the algorithm. SD indicates the current LSDWP of the file. S indicates the set of all the PRCR nodes. The algorithm first calculates the differences between SD and the scan cycles of all available PRCR nodes (step 2-3). Then, from all the PRCR nodes with a scan cycle smaller than SD, the ones with the smallest difference values are selected as the candidates of the destination node (step 4-6). Finally, one of the candidates is randomly chosen as the destination node (step 7). The reason for randomly choosing one node from the node set is to deal with the case where two PRCR nodes have the same scan cycle.

#### Algorithm: Metadata Distribution Algorithm

Input: SD // Storage Duration

S; // PRCR nodes set

Output: node; // selected PRCR node

Step 1: Set diff, nodes;

Step 2: for (each  $i \in s$  & scancycle(i) < SD)

Step 3: diff- SD-scancycle(i)

Step 4: for (each  $j \in S$  & scancycle(i) < SD) {

Step 5: if (SD - scancycle(j)=min(diff))

Step 6: nodes =j ;}

Step 7: node =random (nodes);

Step 8: Return node;

#### Figure 5: Metadata Storage Distribution Algorithm.

## 5. RESULTS

The work is summarized in three steps. As a first step, the registration process of the user will be done with the credentials. Secondly, if the user is authenticated, the User Interface will be able to create a file. The files which are critical will be replicated as two times and the address of the replica will be noted, the other files can be replicated only once. The PRCR node is able to monitor the number of replicas as well as when there is a request from the user from the user interface for the additional replicas it will be considered and a replica is created and at the same time the metadata created and stored in the corresponding file. Thirdly, the generalized reliability model implemented for providing the reliability. The implementation divided into three modules. One is user interface design, second Proactive replica checking and finally, storage prediction is implemented and the snapshots of uploading a file from the user interface, a number of replicas available for the file are shown in Figure 6, Figure 7.



Figure 6: Successful uploading of a file.



Figure 7: Replica Values for the file

## 6. CONCLUSIONS AND FUTURE WORK

In this paper, we presented mainly three contributions. First, a generalized data reliability with multiple replicas is proposed. We have implemented the Proactive Replica Checking for Reliability mechanism by considering the reliability management as a first requirement. Based on that demand, the replication copies will take place according to the request from the user interface, i.e. whether for a specific

data file, either one replica or two replicas are stored. The results show that this method works well and give the better results compared to the existing methods. It reduces the storage space, in turn, reduce the storage cost which is essential for the current cloud storage applications. The performance considerations will be considered as our future work.

## REFERENCES

- [1] Wenhao Li, et.al., "Ensuring Cloud Data Reliability with Minimum Replication by Proactive Replica Checking" in IEEE TRANSACTIONS ON COMPUTERS, VOL. 65, NO. 5, MAY 2016
- [2] W. Li, Y. Yang, J. Chen, and D. Yuan, "A cost-effective mechanism for cloud data reliability management based on proactive replica checking," in Proc. Int. Symp. Cluster, Cloud Grid Comput., 2012, pp. 564-571.
- [3] K. V. Rashmi, N. B. Shah, D. Gu, H. Kuang, D. Borthakur, and K.Ramchandran, "A Hitchhikers guide to fast and efficient data reconstruction in erasure coded data centers," in Proc. Conf. SIGCOMM, 2014, pp. 331342.
- [4] W. Li, Y. Yang, and D. Yuan. 2013, Oct. 11. An energy-efficient data transfer strategy with link rate control for Cloud, Int. J. Auton. Adaptive Commun. Syst. [Online]. Available: <http://www.ict.swin.edu.au/personal/yyang/papers/IJAACS-Li.pdf>
- [5] D. Yuan, Y. Yang, X. Liu, W. Li, L. Cui, M. Xu, and J. Chen, "A highly practical approach towards achieving minimum datasets storage cost in the cloud," IEEE Trans. Parallel Distrib. Syst., vol. 24, no. 6, pp. 1234-1244, Jun. 2013.
- [6] B. Balasubramanian and V. Garg, Fault tolerance in distributed systems using fused data structures, IEEE Trans. Parallel Distrib. Syst., vol. 24, no. 4, pp. 701715, Apr. 2013.
- [7] W. Li, Y. Yang, J. Chen, and D. Yuan, "A cost-effective mechanism for cloud data reliability management based on proactive replica checking," in Proc. Int. Symp. Cluster, Cloud Grid Comput., 2012, pp. 564-571.
- [8] C. Huang, H. Simitci, Y. Xu, A. Ogus, B. Calder, P. Gopalan, J. Li and S.Yekhanin, "Erasure Coding in Windows Azure storage," in Proc. USENIX Annu. Techn. Conf., 2012, pp. 2-13.
- [9] W. Li, Y. Yang, and D. Yuan, "A novel cost-effective dynamic data replication strategy for reliability in cloud data centres," in Proc. Int.Conf. Cloud Green Comput., 2011, pp. 496-502.
- [10] Amazon. (2011). Amazon Simple Storage Service (Amazon S3). [Online]. Available:<http://aws.amazon.com/s3/>, 2011.
- [11] M Armburst, A. Fox, R. Griffith, and A. Joseph, "Above the clouds: A Berkeley View of Cloud Computing", Technical Report No. UCB/EECS-2009-28 by the Department of Electrical Engineering and Computer Science University of California, February 10 2009.
- [12] M Armburst, Armando Fox, Rean Griffith, Anthony D. Joseph, Randy Katz, Andy Konwinski, Gunho Lee, David Patterson, Ariel Rabkin, Ion Stoica, Matei Zaharia, "A view of Cloud Computing, Communications of the ACM", 53(4), 2010, pp.50-58.
- [13] Da-wei Sun, Gui-Ran Chang, Shang-Gao, Li-Zong Jin Xing-ei Wang, " Modeling a Dynamic Data Replication,

Strategy to increase system availability in cloud computing Environments”, Journal of Computer Science and Technology-Springer, 27(2), 2012, pp.256-272.

[14] Aye Chan Ko, Wint Thida Zaw, “ Fault Tolerant Erasure Coded Replication for HDFS based Cloud Storage” , Proc. Fourth IEEE International Conference on Big Data and Cloud Computing, Sydney, Australia, 2014, pp.104-109.

[15] 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, Heraklion, Crete, Greece, September 2010, pp.188-196.

#### Authors Profile:

I, **M.Sravan Kumar** is a PG scholar in the school of Computer Science and Engineering in PDIT, JNTUA. He received his Bachelor’s degree in 2014 from JNTUA University. I am very interested in learning the new technologies and to do various research works and other activities. I have attended various national and international conferences and workshops. My research interests are cloud computing and recommendation systems.

**Mr. K. Ramesh** is currently working as Associate Professor at the school of Computer Science and Engineering in PDIT, JNTUA. He received his master’s degree from Sathyabama University in 2007 and Bachelor’s degree from JNTUA in 2003. He has 10 years of teaching experience and taught various subjects in computer science stream and organized various national conferences and workshops in the organization. His research interests are cloud computing, computer vision and Computer Graphics and knowledge engineering.

**Mr. T. Subramanyam** is currently working as Assistant Professor at the school of Computer Science and Engineering in PDIT, JNTUA. He received his master’s degree from the Central University of Hyderabad in the year of 2013 and Bachelor’s degree from S.V University in the year of 2009. He has 4 years of teaching experience and taught various subjects in computer science stream and organized various national conferences and workshops in the organization. His research interests are machine learning, big data analytics, and knowledge engineering.