Comparative Analysis of Dynamic Replication Strategies in Cloud

Rashmi R. Karandikar
Department of Computer Engineering
St. Vincent Palotti College of Engineering and Technology, Nagpur India

M.B Gudadhe
Department of Computer Engineering
St. Vincent Palotti College of Engineering and Technology, Nagpur India

ABSTRACT
With the emergence of the storage in the cloud, more and more people are moving their data to the cloud. With increased data in the cloud, service providers of cloud face a challenge to ensure maximum data availability and reliability. Data replication is highly employed by commercial large scale cloud storage systems to improve data availability and reliability. In this paper, already existing replication techniques are studied and examined for various parameters like, availability, reliability, storage space consumption, storage cost, bandwidth consumptions, number of replicas, response time. A survey of the different replication strategies is accomplished consolidating the above mentioned parameters. The outcome of such survey will enable the users to determine the replication strategy best suited for their needs.

General Terms
Data Replication Strategies, Cloud Storage Systems.

Keywords
Cloud Computing, Data Availability, Dynamic Replication, Reliability

1. INTRODUCTION
With the rapid progress of technology and the emergence of the internet as a highly reliable system, the organizations have realized the importance of outsourcing of the activities like, computation, processing, etc. as a utility. The services like platforms, network, and storage are outsourced by the organizations heavily as it is perceived as a cost effective option rather than investing heavily on these requirements within. Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources like Networks, servers, storage, applications and services, that can be rapidly provisioned and released, with minimal management effort or service provider interaction.[1] When a Cloud is made available in a pay-as-you-go manner to the general public, it is called as a Public Cloud. The service being sold is called as Utility Computing. Thus, Cloud Computing is inclusive of Software-As-A-Service and Utility Computing [2, 3]. For large organizations it provides lower the infrastructure costs, increased computing power and unlimited storage capacity. The cloud provides features like broad network access, resource pooling and rapid elasticity [3]. However, there are certain disadvantages pertaining to the safety, security and privacy of the data stored. They are network connection dependency, connecting peripherals, ownership of the data, Cost of the data stored, Security 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 [16]. 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 [5]. Smaller organizations, having large web applications have started to move their data to the cloud storage system like Drop Box, Gigaspaces, 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 [5]. Distributed File Systems are a critical component of Cloud Storage Systems. Google File Systems (GFS) [12], Hadoop Distributed File System (HDFS) [13], & Amazon Simple Storage Service (S3) [14] 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 a 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 Necessity for Replication
To increase the data availability, reliability and to improve the access latency in cloud, data replication is widely used. Data replication is a technique in which each logical item of the database has several physical copies each of them located on different machines geographically distributed. [15].Failure of one data node hardly affects the system performance as copy of the same data is available on different locations, therefore improving system availability and reliability. There is distinct a demarcation between data back up and replication. The replication is done for improving the availability, whereas data backup is carried out for disaster recovery. Distributed File 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 [6]. In such distributed file systems, there are different degrees of replication that are conserved in each single of them. The replication levels for more or less commercially available cloud storage systems are, AmazonS3 and Dynamo maintains the reproduction at the detail level, Google File System (GFS)

maintain the return at the chunk level, Hadoop Distributed File System maintain at the cube level. The more are the number of replicas, the higher will be the system availability and dependability. Yet there is also a subsequent growth in the storage space, thus the storage cost which is a major drawback for the user.Data replication algorithm can be split into two broad categories, Static Replication and Dynamic Replication [7]. In static replication, the replication strategy is predefined also the number of replicas and the location is fixed, whereas, dynamic replication automatically creates replicas according to changing access patterns, and location of replicas can be changed dynamically based on capacity of nodes [4].

1.3 Issues for Replication

Dynamic replication strategies have four significant matters that must be spoken. If the wrong choice of data file is done for replication, then it will contribute to unnecessary use of memory space and therefore will increase the memory price. In a similar way there will be the remarkable use of space if too early replication of the data file is performed. Hence it is important to decide which data file is to be replicated and when it is to be replicated. The optimal number of replications to be made must be determined because, an increment in the number of replicas increases the system availability and reliability, but at that place is a break even point beyond which data availability and reliability remain constant. While determining on the optimal number of replicas, the availability and reliability requirement of the user must be ascertained. The price associated with storing these replicas must also be taken, as an increase in the number of replicas will attract a proportionate growth in storage cost. The replicas must be placed in a balanced way such that bandwidth consumption should be minimal; response time should be minimal; and nodes should not be overloaded or minimally loaded. The idea is to place the replicas near to the user so that there is an increase in the throughput of the system.

2. LITERATURE REVIEW

DA-Wei Sun, et al. [4] proposed a dynamic data replication strategy D2RS (Dynamic Data Replication Strategy). The strategy focuses on parameters, System Availability, Number of Replicas & Bandwidth Consumption. The decision for the selection of the data file for replication is based on the theory of temporal locality. The probability that most recently accessed data file will be accessed again in the future, this is called as a theory of temporal locality. The analysis of the data pertaining to access information of the users is carried out. Based on this analysis, the popularity degree and replica factor are calculated. The popularity degree is number of times the block or a file is accessed, within a given time period and is calculated by a time based forgetting function. The replica factor is calculated as ratio of popularity degree and the total number of bytes of data file requested. These two parameters are very crucial in determining the popular data file and also for the decision regarding the replication. When the popularity of a data file passes a dynamic threshold, the replication operation will be triggered. A mathematical relationship is formulated between, system availability requirement and the number of replicas. This relationship reveals the optimal number of new replicas to be created in the system. The number of new replicas to be created at the directly connected data center is calculated based on the total number of new replicas and the replica factor. Replicas will be placed based on access information on directly connected data centers so as meet the system execution rate and bandwidth consumption requirement. The D2RS (Dynamic Data Replication Strategy) strategy is evaluated in Cloudsim environment.Experimental results demonstrate that D2RS(Dynamic Data Replication Strategy) scheme improves data availability, cloud system task successful execution rate, response time, minimizes cloud system bandwidth consumption, and achieves load balance by placing popular data files according to access history . As the probability of block availability increases, higher system byte availability ratio is accomplished with minimal number of replications, yet if the block availability is lowered, then to maintain a higher system byte availability ratio, number of replicas are needed. Also, it is seen that lower the block availability, lower the successful execution rate [4].Qingsong Wei et.al [6] proposed a dynamic replication scheme called CDRM (Cost Effective Dynamic Replication Management) which improves the availability and operation of the system by balancing workload in a cost efficient manner. It is implemented with the help of the Hadoop Distributed File System. (HDFS). The key investigating parameters are Replica Number and Data Availability and its relationship with Replica Number. A mathematical model is proposed to calculate minimum replica number that satisfies the given availability requirement, of a file. If a data node fails and the current replica number is less than the minimum replica number (r_{min}), a new replica will be created on the data node to guarantee the availability requirement. Initially the number of replicas is set to 1 and the current availability of the input file is counted on. If the single replica does not meet this availability then one more replica is added. The availability requirement is gone over again with two numbers of replicas and if it is not satisfied, and so the operation is repeated again till the user specified availability is seen. This system also addresses the subject of how to place the replicas effectively to maximize the load balancing. Replica placement is based on capacity (CPU power, memory, bandwidth etc.) and blocking probability of the nodes in heterogeneous environments. Blocking probability of the node is calculated by using the request arrival rate of the system and the time required to serve the request. The name node maintains the information pertaining to the blocking probability of the entire node by using the data structure B+ tree. Each node calculates the information about its blocking probability and updates it to the name node. Established on the updated information, the name node decides the on which data nodes to place the replicas. The replicas are placed on the data node with the lowest blocking probability.Experimental results demonstrate that CDRM is cost effective and outperforms the default replication scheme of HDFS in terms of execution and load balancing. When popularity is small, the performance of CDRM is much better, than the default replication strategy of HDFS [6].Wenhao Li et al. [9] proposed a cost effective incremental replication strategy (CIR). The CIR is used to handle the data having uncertain reliability requirement. The incremental replication method reduces storage cost simultaneously meeting the data reliability requirement. CIR incrementally creates replicas when current replicas cannot assure the reliability set by the user. Initially, one replica of the data is created and maximum limit of the number of replicas to be created is set to true. Whenever any new data is uploaded by the user or replica creation time point is reached the new replica is created in the system. The replica creation time point indicates that the current number of replicas cannot meet the reliability requirement specified by the user. However, this study does not find the optimal number of replicas nor does it address the placement of the new replicas. Replicas are placed on randomly selected data nodes. As arrangement of new replicas is not taken, the response time reduction and bandwidth consumption are not spoken.

Experimental results indicate that CIR could save upward to two thirds of storage cost as compared to conventional tri-replication scheme. Also CIR occupies less memory space when the storage duration is shorter. [9].

Table-1 presents an overview of how the different replication strategies considered, are speaking the issues like which data to double, when to replicate, how many numbers of replications to be created, where to place these replicas.

Table 1. Issues addressed by strategies

Issues / Strategies	Which data to Replicate	When to Replicate	How many to Replicate	Where to Place
CDRM	Not Addressed	When Access Freq more than Threshold	Mathematical Model to capture the relation between number of replicas and Availability	Replica Placed on data node with lowest blocking probabilit y
CIR	Not Addressed	When Replica Creation time point is reached	No method addressed for how many no of Replicas	Replica Placemen t not addressed
D2RS	Only Popular data	Whose popularity exceeds the threshold	Mathematical Model to capture the relation between number of replicas and Availability	Replica placed on access informati on of the data

3. IMPLEMENTATION OF STRATEGIES IN HDFS

Hadoop provides a distributed file system that is designed to store very large data sets with high reliability. It streams the data sets at high bandwidth to user applications. It stores the file system metadata on a dedicated server called as Name Node and application data is stored on another server called as DataNode. HDFS client is coded library using which the user application access the file systems. If an HDFS client wishes to read a file, it first contacts the name node for the locations of the data blocks and then it reads the contents of the data node nearest to the client. An application can specify the number of replicas in a file and replication factor is configurable for a file in HDFS.

CDRM (Cost Effective Dynamic Replication)

When a user submits a file along with the availability requirement to the HDFSclient, it is divided into multiple blocks and stored in HDFS. Initially, only one replica of each block is maintained. The availability of a file is calculated which is based on parameters like block locations, current number of blocks, number of replicas, network bandwidth, etc. In this strategy an assumption is made that a block of a file is not available if a node hosting that file is not available. The probability of the block unavailable is given by

probBlockNotAvailable =
probBlockNotAvailable x nodefaliureProbability

Where, **probBlockNotAvailable** depends upon number of replicas. As number of replica increases, the **probBlockNotAvailable** decreases. The default value of **probBlockNotAvailable** is assumed to be 1

So,

probBlockNotAvailable = 1 X (**nodeFaliureProbability**)^r

Where, r = Current number of replicas.

The file availability is calculated as

$$\text{FileAvailability} = 1 - \text{FaliureProbability} \quad (1)$$

$$\sum_{b=0}^{n}[\sum_{c=0}^{n}[[BlockFaliureProbCombination[0] * BlockFaliureProbCombination * \ldots\ldots BlockFaliureProbCombination[n]] * (-1)^r]$$

Where FaliureProbability =

Where b varies from 0 to n and is equal to the number of blocks generated for a file. c varies from o to n and is equal to the number of combinations for a particular block not being available for different nodes and r is the count of replication. If the calculated availability is meeting the given availability requirement, then file is not replicated. If the requirement is not met, and if the replication count has not reached its maximum limit of three, then the replica is added to the system. Below is given the pseudocode for CDRM [6].

1. Input: - Input file ,requiredAvailability, blockSize, startReplicationfactor = 1, nodeFailure Probability.
2. While (true)

   Find out the current number of replicas of that file
3. Calculate the availability of the input file for the current number of replicas
4. If (Calculated Availability >= RequiredAvaialability)

   Then

   File is meeting user specified availability requirement

   Break;

   Else
5. If (Replication < max_limit)
6. Then

Fig-1 CDRM- Cost Effective Dynamic Replication Strategy Psuedocode

CIR (Cost Effective Incremental Replication)

In CIR [8] the automatic replication is controlled by a threshold, where the threshold is a time duration after which the file should be checked for replication. CIR employs a mechanism for continuously managing the replication of the files added to the cluster.

When the ReplicaCreationThreshold exceeds, then the current number of replicas is checked whether they can meet the user specified availability requirement or not. If not, then replica is added and this process continues until the maximum limit (maximum of 3) of replica is reached. Unlike CDRM which does not check the file periodically and hence leading to difficulty in tracking the loss of any block of the file, CIR periodically monitors the availability and replica of the file due to which potential risk to the file can be detected easily and minimized.

D2RS (Dynamic Data Replication Scheme)

In D2RS [4] the automatic replication of files is carried out on the basis of popularity. When the popularity count of the file exceeds the popularity Threshold limit, then file will be checked for replication. In this strategy initially only one replica is maintained until the popularity crosses the threshold limit because of which the storage space consumed is considerably less compared to the other two strategies.

1. Input :- Input file ,ReplicaCreationThreshold

   NodeFaliureProbability, Blocksize, RequiredAvalability, StartReplicationFactor
2. For each time window 'T'
3. Check all the files on the specified path.

   Add all the files crossing ReplicaCreationThreshold limit in the list.
4. For each file in the list
5. Get files replication count
6. Calculate its availability

   If (Calculate availability >= Required Availability)
7. File is meeting availability Requirement

Fig-2 Cost Effective Incremental Replication Strategy Pseudocode

1. Input: - Input file, blocksize,popularityThreshold, startReplicationFactor, requiredAvailability,nodeFaliureProbability.
2. For each file in the list

   //Check Whether the File Exceeds Popularity Threshold
3. If (popularity_Count >= popularityThreshold

   //Check the availability of the File
4. Get files replication count

   If (Calculate availability >= Required Availability)
5. File is meeting availabilityRequirement
6. Else
7. If (Replication < max_limit)
8. Replica = Replica +1

Fig-3 Dynamic Data Replication Management Scheme Pseudocode

4. EVALUATIONAND COMPARISION

The test platform for the evaluation was built on a cluster of one name node and two slave nodes of commodity computer. Each node has an Intel Pentium IV CPU of 2.8 GHz, 1 GB or 2 GB memory, 80GB or 160 GB SATA disk. The operating system is UBUNTU 14.4. Hadoop version is 16.0.For the evaluation the datasets of different sizes were used. Each file is stored in fixed size (1 kB) storage unit called block. Blocks of the same data file are scattered along different storage unit called block. At the very beginning the number of replica of each data file is one. For placement of replicas the default HDFS block placement policy is used. In this section the following parameters are analyzed for the three replication strategies.

A. Availability

From the equation (1) the availability for the files is calculated by all the three strategies. Initially the number of replica is set to one. The node failure probability is set to 0.2 and the required availability is set to 0.992. After evaluation it was found out that,

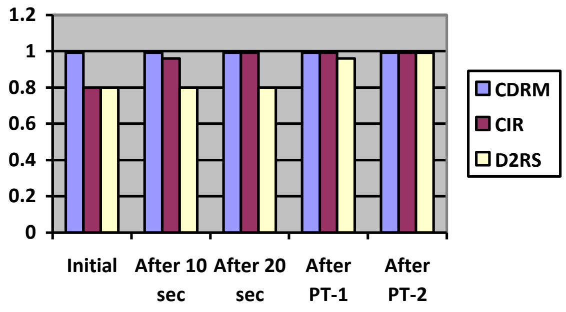
-For replica number = 1, the file availability comes out around 0.8 for all the three strategies.

-For replica number = 2, the file availability comes out around 0.96.

- For replica number = 3, the file availability comes out around 0.992.

Table 2. Availability comparison of three strategies for a 100 MB file with maximum replication of 3

Condition	CDRM	CIR	D2RS
Initial	0.992	0.8	0.8
After 10 seconds	0.992	0.96	0.8
After 10 seconds	0.992	0.992	0.8
After Popularity Threshold 1	0.992	0.992	0.96
After Popularity Threshold 2	0.992	0.992	0.992

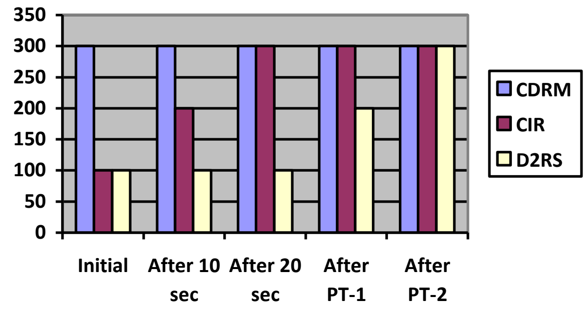


From the above graph it is observed that for CDRM since three replicas are created at the onset, the availability consistently remains constant at 0.992.But CIR incrementally creates replicas only after meeting replicaCreationThreshold. The default value is set to 10 seconds. Initially for one replica the availability value is set to 0.8.If the file is 10 seconds older then second replica is added and availability increases to 0.96. Again after 10 seconds the availability finally reaches to 0.992 when the maximum limit of 3 replicas is reached.In case of D2RS, the availability remains constant 0.8 until the popularityThreshold is not exceeded. popularityThreshold is defined as number of times the file is accessed by the user. As soon as the popularityThreshold 1 is reached, the second replica is added leading to availability of 0.96. As popularityThreshold 2 is reached the third replica is added, and availability of 0.992 is met.From the graph it is found that, CDRM is meeting the users specified availability requirement is less time, as compared to that taken by CIR and D2RS, therefore CDRM is better in terms of availability as compared to CIR and D2RS.

B. Storage Space

Table 3. Storage comparison of three strategies for a 100 MB file with maximum replication of 3

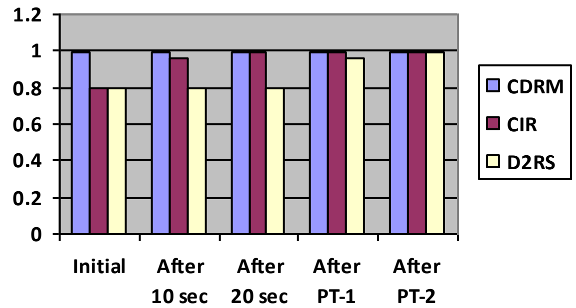
Condition	CDRM	CIR	D2RS
Initial	300	100	100
After 10 seconds	300	200	100
After 10 seconds	300	300	100
After Popularity Threshold 1	300	300	200
After Popularity Threshold 2	300	300	300



In the above graph a file of 100 MB size is considered and it is observed that CDRM initially consumes 300 MB storage space as it creates three replicas of the file. In case of CIR after replicaCreationThreshold is met and the number of replicas are created then the storage space used sequentially increases after every 10 seconds. For D2RS only 100 MB is utilized as only one replica is created. Until the popularityThreshold is met no replica is created and only 100 MB is utilized, therefore it is using the storage space more effectively as compared with CDRM and CIR

Table 4. Bandwidth Comparison of Three Strategies for a 100 MB file with maximum replication of 3

Condition	*CDRM*	*CIR*	*D2RS*
Initial	300	100	100
After 10 seconds	0	100	0
After 10 seconds	0	100	0
After Popularity Threshold 1	0	0	100
After Popularity Threshold 2	0	0	100



In the above graph a file of 100 MB size is considered and it is observed that CDRM initially consumes 300 MB bandwidth and as it creates three replicas of the file. In case of CIR after replicaCreationThreshold is met and the number of replicas are created. But as only one replica is created after every 10 seconds then therefore bandwidth consumption is restricted to 100 MB. For D2RS only 100 MB is utilized as only one replica is created. Until the popularityThreshold is met no replica is created and only 100 MB is utilized, therefore CIR and D2RS use bandwidth more effectively as compared with CDRM.

5. CONCLUSIONS & FUTURE WORK

In this paper, three dynamic replication techniques are surveyed and compared for parameters like availability, storage space, and bandwidth consumption. These replication strategies were implemented in HDFS and experimental results demonstrate that CDRM meets the user specified availability requirement much faster than CIR and D2RS. The D2RS replication strategy, consciously utilizes the storage space by creating the replicas of only popular file, however CDRM creates the replicas of all the files and hence having a higher storage space consumption.It is expected that such a comparison may give an insight into the gains and drawbacks of these strategies. The observed drawbacks can be investigated and an improvement can be proposed in the performanceMost of the data used in real time is updatable therefore; the consistency of the replica stored is an important parameter which is not addressed by any of the above strategies. This parameter could be an important research area in future. It is also observed that very few strategies are considering the failure rate of the storage units in cloud. Detailed experimentation is needed to evaluate the effect of the disk failure on the availability and reliability of the data stored in cloud. Most of the replication strategies are compared with the default three replica strategies. Further work can be carried out in the area of reducing the user waiting time, speeding up data access, and increasing the data availability.

6. REFERENCES

[1] A Bagha, V. Madesetti, “Cloud Computing- A Hands-on- Approach “2014, pp 15-19.

[2] 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.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] Wenhao-Li, Yun Yang, Dong-Yuan, “Ensuring Cloud Data Reliability, with minimum replication by proactive replica checking”, 2015 IEEE transactions on computers and Manuscript ID.

[9] Wenhao-Li, Yun Yan, Dong Yuan, “A Novel Cost Effective Dynamic Data Replication Strategy for Reliability in Cloud Data Centers”, Ninth IEEE International Conference on Dependable, Autonomic and Secure Computing, Sydney, Australia, December 2011,pp.496-501.

[10] Mohammed K. Hussein, Mohammed K. Mousa, “A lightweight Data Replication for Cloud Data Centers Environment” , International Journal of Innovative Research in Computer and Communications Engineering, 2(1), 2014, pp.2392-2400.

[11] Tehmina Amjad, Muhammad Sher, Ali Daud, “A survey of Dynamic Replication strategies for improving data availability in data grids”, Future Generation Computer Systems, 28, 2012, pp.337-349.

[12] S. Ghemawat, H. Gobioff, and S. Leung, "The Google file system," in ACM Symposium on Operating Systems Principles, pp. 29-43, 2003.

[13] K. Shvachko, H. Kuang, S. Radia, R. Chansler, “ The Hadoop Distributed File System”, 26th International Symposium on Mass Storage Systems and Technologies Lake Tahoe, Nevada, USA, May 2010.

[14] Amazon. (2011). Amazon simple storage service (Amazon S3). Available: http://aws.amazon.com/s3/

[15] R. Ranjana, T.P.jayalakshmi, R. Srinivasan “On Replication Strategies in Cloud for Data Intensive Cloud Applications “, International Journal of Computer Science and Information Technology, Vol. 6 (3), 2015, pp 2479-2484.

[16] Rashmi Karandikar, Manish Gudadhe “Analytical Survey of Dynamic Replication Strategies in Cloud”, Proc. IJCA-National Conference on Recent Trends in Computer Science and Information Technology 2016 (1), pp1-5