International Journal of Applied Engineering Research ISSN 0973-4562 Volume 13, Number 22 (2018) pp. 15926-15932

A Comprehensive Survey on Data Replication Techniques in Cloud Storage Systems

Suji Gopinath

Research Scholar, University of Kerala, Kerala, India.
Assistant Professor, Department of Computer Applications, N.S.S College Rajakumari, Idukki, Kerala, India.
Orcid Id: 0000-0002-5653-7611

Elizabeth Sherly

Professor, Indian Institute of Information Technology and Management-Kerala
Trivandrum, Kerala, India.

Abstract

Cloud storage is a service model for data storage, which is located in the network on large number of servers, where data is maintained, managed and backed-up remotely. Data replication is a technique used by large-scale cloud storage systems to ensure high availability, reliability and consistency. In this paper, a detailed study and survey of the various data replication techniques in cloud storage systems are presented. Data replication techniques falls into categories named static and dynamic replication. In a static data replication model, the number of replicas to be created and the node to place the replica is decided statically at the time of cloud storage system setup. However, in dynamic replication model, creation and placement of data is done dynamically according to changing environment and user access pattern. The review is completed with classification and comparison of the reviewed mechanisms and some related issues and some suggestions to solve it are mapped out.

Keywords: Cloud storage, Data replication, Static replication, Dynamic replication

INTRODUCTION

Cloud computing[1] is an internet-based computing platform that provides on-demand access to the virtualized computing resources like servers, storage, applications, software, computer networks, services etc. It has became a highly demanded service due to its ability to provide high computing power, high performance, scalability, availability and accessibility with cheap cost of services. The virtual resources, both hardware and software are provided as on-demand services to the end-users over Internet namely Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a service (IaaS) [2]. The capability provided in the Software as a Service (SaaS) architecture to the consumer is to use the software applications provided by a cloud service provider which is running on a cloud infrastructure. In Platform as a Service (PaaS) architecture, an application delivery platform by the service provider is provided, which enables the consumer to deploy end-user applications created using programming languages, libraries, services, and tools supported. . Infrastructure as a service (IaaS) architecture allows the consumers to provision processing, storage, networks, and other fundamental computing resources which enables the deployment of operating systems and applications.

Cloud storage is a powerful IaaS model in which customer's data is stored, managed remotely and made available to customers across Internet. Cloud storage systems need to meet several essential requirements like high availability, reliability, performance, replication and data consistency to maintain customers' data. In cloud storage systems, files are split into multiple blocks and stored in different data nodes across the distributed network. Unreliable network connectivity, erratic node failure and limited bandwidth will create challenges in efficient data sharing. The probability of data node failure is high in a cloud storage system which will affect the availability of data. If a data node which stores a block of the file fails, then the entire file will be unavailable. For data intensive applications like scientific applications needs the data to be available round the clock. Large-scale cloud storage systems use data replication to ensure high availability, reliability and consistency of data stored in it. Data replication is a commonly used strategy by which same data is stored in multiple storage devices. The chances of at least one copy of data being available at any time increases when multiple copies of data are stored in different data nodes.

The paper is further organized as follows, Section 2 discusses the different data replication approaches used for cloud storage and the significant contributions of various researchers towards data replication. Section 4 presents the comparisons of the reviewed strategies. In section 5, we point out our future research plan in data replication together with the concluding remarks.

DATA REPLICATION APPROACHES

Data replication is a major technique used in distributed systems to meet the challenges of high availability and to improve data access performances. Data replication increases data availability and reliability, reduces user waiting time, and minimizes cloud system bandwidth consumption, increases fault tolerance and improves scalability. When data is

replicated, copies of data files are created at many different data nodes in the cloud storage systems. If one data node fails, a replica of the data will be available in a different node to process the request, thereby giving uninterrupted service.

Data replication falls into two categories namely static replication [18, 4, 19] and dynamic replication. In a static data replication model, the number of replicas to be created and the node to place the replica is defined statically at the time of cloud system setup. On the other hand, dynamic data replication can adapt changes according to user requests, storage capability and bandwidth. It can automatically create and deletes replica according to changing environment. The static and dynamic replication algorithms can be further classified as centralized or distributed algorithms. There have been lots of researches done to resolve the issues of data replication in cloud storage systems.

Static Data Replication Approaches

Static data replication strategies follow a deterministic approach, where the number of replicas to be created and the node to place the replica is well defined and pre-determined. It statistically replicates data on randomly chosen nodes for a fixed number of times. This method provides faster response, high availability and high efficiency. These strategies are easy to implement, but not often used since it cannot adapt to changes in user request, storage capacity and bandwidth.

Ghemawat et al. (2003) [3] designed a scalable distributed file system called Google File System for data intensive applications. This file system provides efficient, reliable access to large data set using large clusters of inexpensive hardware. GFS implements a static distributed data replication algorithm for Google Cloud. In this the replicas in multiple chunk servers are dynamically maintained. The replicas of a chunk are spread across racks and a data chunk is dynamically replicated when the number of replicas falls below the limit specified by the user. The new chunk replicas are placed in the servers having below-average disk space utilization. The limitation of this approach is that a fixed replica number is used for all files which may not be the best solution for data replication.

Hadoop Distributed File System[4], the storage component of Apache Hadoop follows a static distributed replication policy to provide availability and reliability of data. The replication factor and block size for each file is configured at the time of file creation. It follows a triplication policy for replicating data blocks across Hadoop cluster and a rack-aware replication placement policy to place the replicas. The placement of these replicas is done such that two replicas are stored in two separate nodes in the same local rack and one in a separate remote rack. This replica placement policy cuts the inter-rack write traffic, which generally improves writing performance. The rack-aware replication placement strategy improves data reliability, availability, and network bandwidth utilization. The limitation of this approach is that access behavior is not taken into consideration for replicating data.

Amazon Dynamo [5] is a static decentralized system with minimal manual administration. It replicates each data item at a fixed number of physically distinct nodes. Storage nodes are added and removed from Dynamo without requiring any manual partitioning or redistribution. Dynamo deals with load balancing by assuming the uniform distribution of popular data items among nodes through partitioning. Data consistency is handled based on vector clocks and a quorum system approach with a coordinator for each data key.

Hassan et al. (2009) [6] proposed a multi-objective optimization approach for replica management in large-scale cloud storage cluster using evolutionary method. Two novel algorithms named multi-objective Evolutionary (MOE) algorithm and multi-objective Randomized Greedy (MORG) algorithm is presented for deciding the number of replicas and their placement within the overlay. MOE improves not only improves storage, latency, and availability but also search for solutions to optimize these parameters. The framework was inspired by the evolutionary computing framework and each solution was presented as a chromosome.

Cidon et al. (2013) [7], presented a simple, general-purpose and scalable replication scheme called MinCopysets which de-randomizes data replication in order to achieve better data durability properties. It allows system designers to use randomized node selection for data distribution to get the benefits of parallelization and load balancing. In this method the servers are divided statically into replication groups (each server belongs to a single group). When a chunk has to be replicated, a primary node is selected randomly for its first replica, and the other replicas are stored deterministically on secondary nodes, which are the other nodes of the primary node's replication group.

Long et al., (2014) [8], developed a Multi-objective Optimized Replication Management (MORM) strategy which is an offline optimization framework for replica management. The decision of replication factor and replication layout is done using the improved artificial immune algorithm. Five objectives were considered such as mean file unavailability, mean service time, load variance, energy consumption and latency. Mathematical models were formulated to describe these objectives while considering the size, access rate of each data file, failure probability, transfer rate and capacity of each data node. A suitable number of replicas are maintained for each data file to achieve the optimal objective value and replicas are placed among data node with respect to the five objectives. MORM applies to the scenario where access statistics is fixed and therefore the replication strategy is computed only once. But it is not suitable when files arrive in the storage dynamically and continuously.

The replication and the data placement strategies in static replication techniques are pre-determined and well-defined. The static replication strategy keeps the number of active data replicas at the maximum with a random data placement policy. The creation of maximum number of replicas may guarantees better performance, but at a high operational cost. A detailed comparison of the reviewed static data replications techniques are summarized in Table I.

Table 1: Comparison of static replication techniques studied in the literature

Approach	GFS[3]	HDFS[4]	Amazon Dynamo[5]	MOE[6]	MinCopysets[7]	MORM[8]
Year	2003	2007	2007	2009	2013	2014
Replication pattern	static	static	static	static	Static	static
Improved availability	Yes	yes	Yes	No	Yes	Yes
Reliability	Yes	yes	yes	Yes	Yes	Yes
Reduced response time	Yes	yes	yes	Yes	No	Yes
Bandwidth consumption	High	High	High	High	High	High
load balancing	Yes	Yes	Yes	No	No	Yes
Optimal number of replicas	No	No	No	No	No	Yes
Replication cost consideration	No	No	No	No	Yes	No
Reduced access cost	No	No	No	No	No	Yes
Storage cost	High	High	High	High	Low	High
Energy reduction	No	No	No	No	No	Yes

Dynamic Data Replication Approaches

Dynamic data replication approaches can dynamically create and delete replicas based on the changes in user access pattern, storage capability and bandwidth. . All data in the cloud storage may not follow a common access pattern. Some data will be frequently accessed and some may be least accessed. A dynamic replication policy is needed in such scenario where the replicas for each data is decided based on the access popularity of data. Dynamic approaches can reduce storage space utilization and cost by maintaining the replica for each data item intelligently. Dynamic data replication policies need to make some critical decisions: which data need to be replicated and when, how many replicas need to be created and where to place the new replicas.

Wei et al. (2010) [9] proposed CDRM, a Cost-effective Dynamic Replication strategy for large-scale cloud storage systems which concentrates on capturing the relationship between availability and replica factor. In this work the popularity of a data file is calculated to create replica for the data file. After finding the popular file, lower bound on replica reference number is determined that satisfies the availability requirement. The new replica is placed in a suitable node considering the blocking probability and capacity of the nodes. CDRM provides cost-effective availability and improves performance and load balancing in the cloud storage by maintaining a minimal replica number for a given availability condition. This strategy dynamically reallocates workloads among data nodes by regulating replica number and location according to workload changing and node capacity.

Li et al. (2011) [10] presented Cost-effective Incremental Replication (CIR), a novel cost-effective dynamic data replication strategy for cloud data centres. It is a data reliability strategy for cloud based applications for cost-effectively managing the data reliability issue in a data centre. The idea of CIR is to use the minimum number of replicas while meeting the data reliability requirement. An incremental replication method is implemented in CIR for calculating the replica creation time point which indicates that the current number of replicas cannot assure the data reliability requirement any longer and a new replica should be created. Initially, the minimum data replica number is set to 1 by default. When time goes by, more replicas are incrementally created at the set time points to maintain the reliability assurance. The result of the evaluation indicates that CIR strategy significantly reduces the storage cost of the entire storage system by reducing the number of replicas whilst meeting the reliability requirement.

Ananthanarayanan et al. (2011) [11] proposed an off-line system called Scarlett, which periodically replicates data blocks based on its popularity and spread them out to avoid hotspots, with minimal interference to running jobs. Scarlet adopts a proactive replication scheme where popular files to be replicated are identified using prediction method based on the historical usage and jobs submitted for execution. After finding the popular files, replication factor for these files are increased and the replicas are placed in such a way that hotspots are minimized in the cluster and minimal interference to the cross-rack network traffic. This method uses the concept of aging for replicas to give space for new replicas which may cause loss of some files.

Abad et al. (2011) [12] proposed a distributed adaptive data replication algorithm named DARE, which replicates the data files dynamically to increase data locality. The number of replicas to be created for each file and the node to place the replica is determined based on the probabilistic sampling and a competitive aging algorithm independently at each node. Replicas created by DARE are first-order replicas which also contribute to the increasing availability of data in the case of failures. It creates replicas without consuming extra computation resources and network by taking advantage of remote data retrievals and selects a subset of the data to be inserted into the file system. A probabilistic approach is also applied to the aging mechanism which evicts files with decreasing popularity. The replication decision is based on

probability which does not consider the trends of data utilization.

Kaushik et al. (2011) [13] proposed a predictive data replication policy named GreenHDFS, an energy-conserving variant of the Hadoop Distributed File System which uses supervised machine learning methodology to learn the correlation between the directory hierarchy and the file attributes such as file's lifespan, heat and size to guide novel predictive file zone placement, migration, and replication policies. The predictive models are used at the time of file creation to predict file attributes. GreenHDFS proactively create and delete replicas based on the file heat predictions. Replicas are created for the hot files and the replica of cold file gets deleted. This method fails to consider the management of cold data efficiently and the directory structure is an important factor in the anticipation for file accesses.

Cheng et al. (2012) [14] proposed an elastic replica management system for HDFS named ERMS, which provides an active/standby model for the storage of data in HDFS. The real-time data is classified into hot or cold by using the complex event processing engine and replica is created dynamically based on this classification. The replica for the hot data is increased and the extra replicas for cold data are cleaned up and erasure coding applied to it. ERMS classifies the storage nodes into active nodes and standby nodes. The new replicas for the hot data are placed in the standby nodes in the assumption that standby nodes may be better than the active nodes when active nodes are heavily used. When replica for the cold data is decreased, ERMS does not need to rebalance the replicas if they are located in standby nodes. After removing all the data in a standby node, that node is shut down for energy saving. ERMS dynamically adapts to the changes in data access patterns and data popularity.

Sun et al. (2010) [15] proposed a dynamic data replication strategy named D2RS, where a mathematical model is formulated which describes the relationship between the system availability and number of replicas while considering the factors like size, access time and failure probability of each data. A weight is set to different accessed data by analyzing the access histories. The popular data is set a big weight. The replication operation is triggered when the popularity of the data passes a dynamic threshold value. The suitable number of replicas is calculated to meet a reasonable system byte effective rate requirement. The new replicas are placed in data nodes by considering the access information of directly connected data centres and are accomplished in a balanced way. The result of the evaluation shows that the proposed strategy increases the data availability and minimizes the cloud system bandwidth consumption.

Kousiouris et al. (2013) [16] proposed a proactive data management framework for Hadoop clusters which is based on predictive data activity patterns. The method will predict the future data demand in the HDFS cluster using Fourier series analysis [12] and dynamically calculates the replication factor to meet the availability requirement, improve performance and minimize the storage utilization. This framework keeps the statistics of data access usage and feed them in a service oriented time series prediction component. The prediction component creates prediction models based on the previous access values and identifies the data items that are expected to be more popular in the future. The replica for the future anticipated popular data is increased and the popularity declining data is decreased. In this method file is classified only in limited replication scenarios.

Boru et. al (2015) [17] presented an energy-efficient data replication scheme for cloud computing data centers which takes into account power consumption and bandwidth required for data access. In this scheme a three tier fat tree datacenter topology is used which consists of a Central database, local datacenter databases hosted by each data center and rack-level databases hosted by each racks. The Central DB hosts all the data required by the cloud applications. The Datacenter DB is used to replicate the most frequently used data items from the Central DB and Rack DB is used for subsequent replication from the Datacenter DB. A module called replica manager (RM) located at the Central DB, periodically analyze data access statistics to identify which data items are most suitable for replication and at which replication sites. The usage of access and update statistics helps in decreasing the energy and bandwidth consumption of the system.

Bui et al. (2016) [18] proposed an adaptive replication management (ARM) in HDFS based on supervised learning to provide high availability for the data in HDFS by enhancing the data locality metric. This method replicate the data files based on the predictive analysis. A complexity reduction method for prediction technique was adopted for both hyper-parameter learning and training phases to increase the effectiveness of the prediction. The popularity of each data file is predicted and high potential files are replicated while erasure code is applied to the low potential files. The new replicas are placed on low utilization nodes which have low blocking probability, to redirect the tasks to these idle nodes and balance the computation. The evaluation result shows that this strategy improves the availability while keeping the reliability.

Qu et al. (2016) [19] proposed a dynamic replication strategy (DRS) based on Markov Model for HDFS which consists of a dynamic replica adjustment strategy (DARS) and a homogeneous replica placement strategy (HRPS). In this method a transition probability matrix is first constructed based on the accessing of files in a time period. DRS then compute the stationary probability distribution depending on the initial distribution and the transition probability matrix and find the number of replicas that should be assigned for each file. Using the results obtained data is classified as hot or cold. Extra replica is created for hot data and replica of cold data is deleted. The homogeneous replica placement strategy is used to distribute the replicas across HDFS cluster, which considers the relationship between different kinds of data to place the relevant data on the same node or rack in order to reduce the data transmission time and bandwidth consumption between the nodes or racks. This method is not considering the effective management of cold data resulting in a probability of data loss.

S. Gopinath and E. Sherly (2018) [20] proposed a weighted dynamic data replication policy for cloud data storage where the data is replicated dynamically by classifying it to hot, warm or cold data. A weight is assigned to each data item based on its access popularity in the system. A popularity index for each data item is calculated based on its access count, weight and current replication factor. Then the data is classified to hot, warm or cold considering its popularity index, a calculated threshold value, weight. Replica factor for hot and warm data is calculated dynamically. A minimum replica factor is set to cold data item and erasure coding is applied to guard against data loss. The result of the evaluation shows that this strategy maintains an optimal replication factor which is adequate to ensure the reliability and availability of the data. It also reduces the storage cost by reducing the storage space consumption.

Dynamic data replication techniques makes intelligent decision in creating replicas for data dynamically based on the changes in the current environment. Most dynamic replication strategies create a new replica of the popular data based on the user access frequency and the placement of the replicas are done dynamically. Dynamic replication approaches can thus reduce storage space utilization and cost by wisely maintaining the replica for each data item. A detailed comparison of the reviewed dynamic data replications techniques are summarized in Table II.

Table 1: Comparison of dynamic replication techniques

Approach	CDRM [9]	CIR [10]	Scarlett [11]	DARE [12]	GreenHDFS [13]	ERMS [14]	D2RS [15]	Kousiouris [16]	Boru [17]	ARM [18]	DRS [19]	WDRM [20]
Year	2010	2011	2011	2011	2011	2012	2012	2013	2015	2016	2016	2018
Replication pattern	Dynamic	Dynamic	Dynamic	Dynamic	Dynamic	Dynamic	Dynamic	Dynamic	Dynamic	Dynamic	Dynamic	Dynamic
Improved availability	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Reliability	No	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes
Reduced response time	No	No	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes
Bandwidth consumption	High	Low	High	Low	High	High	Low	High	Low	High	Low	Low
Load balancing	Yes	No	Yes	Yes	Yes	Yes	No	Yes	No	No	Yes	Yes
Optimal number of replicas	Yes	No	No	No	No	Yes	No	Yes	No	Yes	No	Yes
Replication cost consideration	No	Yes	No	No	No	Yes	Yes	Yes	No	Yes	No	Yes
Reduced access cost	Yes	No	No	No	No	No	No	No	Yes	No	Yes	Yes
Storage cost	High	Low	High	High	High	Low	High	Low	High	Low	High	Low
Energy reduction	No	No	No	No	Yes	No	No	No	Yes	No	No	No

FUTURE DIRECTIONS

From this literature survey, it is clear that there are still lots of works to be done in the field of cloud data storage. Some of the important features to be considered while replicating data are discussed in this section.

The replication decision making is an important feature. The replication decision can be centralized or distributed. In centralized systems there is a chance of bottleneck in case there is more than an average load on the network and in case of distributes systems there is a chance of unnecessary replications.

Almost all the replication strategies are found to increase the availability of data and reduce the response time. Some strategies maintain optimal bandwidth consumption, but it is not considered by most of the replication strategies. The increased availability of data is sometimes at the cost of more bandwidth consumption.

The storage space consumed by a replication strategy is another important feature to be considered. Some of the strategies ensured reduced storage space consumption by maintaining an optimal number of replicas.

It is found that there is no single strategy that addresses all issues involved in data replication. Some strategies focused on

providing availability, reliability, fault tolerance and load balancing, while some others focused on preserving the network bandwidth. A comprehensive technique need to be designed which will consider all the parameters needed for a better data replication.

Most of the strategies used simulation to evaluate the algorithms. In future, these systems need to be prototyped and evaluated in real world scenarios to provide a realistic evaluation of the strategies.

Also most of the strategies considered that the data is read only, but in reality data is not always read only. A replication strategy which can deal with updatable data as well is a need for the hour.

A replication strategy that supports replica management in terms of replica creation, deletion and placement, which reduces both job execution time and network traffic, can be considered as a future research.

CONCLUSIONS

Replicating data among different nodes in a cloud-storage system ensures reliability and high availability of data. Among the static and dynamic data replication methods, dynamic data replication is powerful, since it considers the change in pattern of data access. In this paper, several proposed approaches in each method is reviewed and compared. There are several issues related to data replication which need to address while replicating data such as availability, reliability, scalability, fault tolerance, reduced response time, faster data access, optimal number of replicas, load balancing, band width consumption etc. From this review, it has been seen that there is not a single approach that addresses all the issues related to data replication. Therefore, developing a data replication strategy for cloud storage that considers all important data replication parameters is a need for the hour.

REFERENCES

[1] Buyya, R., et al., Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility. Future Gener. Comput. Syst., 2009. **25**(6): p. 599-616.

[2] Mell, P., & Grance, T. (2011). The NIST definition of cloud computing.

[3] Ghemawat, Sanjay, Howard Gobioff, and Shun-Tak Leung. (2003) "The Google file system." *19th ACM Symposium on Operating Systems Principles***37 (5)**: 29-43.

[4] White, T. (2012). Hadoop: The definitive guide. "O'Reilly Media, Inc.".

[5] DeCandia, G., Hastorun, D., Jampani, M., Kakulapati, G., Lakshman, A., Pilchin, A., ... & Vogels, W. (2007, October). Dynamo: amazon's highly available key-value store. In *ACM SIGOPS operating systems review* (Vol. 41, No. 6, pp. 205-220). ACM.

[6] Hassan, O. A. H., Ramaswamy, L., Miller, J., Rasheed, K., & Canfield, E. R. (2008, November). Replication in overlay networks: A multi-objective optimization approach. In *International Conference on Collaborative Computing: Networking, Applications and Worksharing* (pp. 512-528). Springer, Berlin, Heidelberg.

[7] 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)*.

[8] Long, S. Q., Zhao, Y. L., & Chen, W. (2014). MORM: A Multi-objective Optimized Replication Management strategy for cloud storage cluster. *Journal of Systems Architecture*, *60*(2), 234-244.

[9] Wei, Qingsong, Bharadwaj Veeravalli, Bozhao Gong, Lingfang Zeng, and Dan Feng. (2010) "CDRM: A cost-effective dynamic replication management scheme for cloud storage cluster.", in *Cluster Computing (CLUSTER), 2010 IEEE International Conference on* , IEEE : 188-196.

[10] Li, Wenhao, Yun Yang, and Dong Yuan. (2011) "A novel cost-effective dynamic data replication strategy for reliability in cloud data centres.", in *Dependable, autonomic and secure computing (DASC), 2011 IEEE ninth international conference on*, IEEE : 496-502.

[11] Ananthanarayanan, Ganesh, Sameer Agarwal, Srikanth Kandula, Albert Greenberg, Ion Stoica, Duke Harlan, and Ed Harris. (2011) "Scarlett: coping with skewed content popularity in mapreduce clusters.", in *Proceedings of the sixth conference on Computer systems*, ACM : 287-300.

[12] Abad, Cristina L., Yi Lu, and Roy H. Campbell. (2011) "DARE: Adaptive data replication for efficient cluster scheduling.", in *Cluster Computing (CLUSTER), 2011 IEEE International Conference on*, IEEE : 159-168.

[13] Kaushik, R. T., Abdelzaher, T., Egashira, R., & Nahrstedt, K. (2011, July). Predictive data and energy management in GreenHDFS. In *Green Computing Conference and Workshops (IGCC), 2011 International* (pp. 1-9). IEEE.

[14] Cheng, Zhendong, Zhongzhi Luan, You Meng, Yijing Xu, Depei Qian, Alain Roy, Ning Zhang, and Gang Guan. (2012) "Erms: An elastic replication management system for hdfs.", in *Cluster Computing Workshops (CLUSTER WORKSHOPS), 2012 IEEE International Conference on*, IEEE : 32-40.

[15] Sun, Da-Wei, Gui-Ran Chang, Shang Gao, Li-Zhong Jin, and Xing-Wei Wang. (2012) "Modeling a dynamic data replication strategy to increase system availability in cloud computing environments." *Journal of computer science and technology* **27 (2)**: 256-272.

[16] Kousiouris, G., Vafiadis, G., & Varvarigou, T. (2013) "Enabling proactive data management in virtualized hadoop clusters based on predicted data activity patterns." In *P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC), Eighth International Conference on* IEEE: 1-8

[17] Boru, D., Kliazovich, D., Granelli, F., Bouvry, P., & Zomaya, A. Y. (2015). "Energy-efficient data replication in cloud computing datacenters." *Cluster computing*, *18*(1), 385-402.

[18] Bui, Dinh-Mao, Shujaat Hussain, Eui-Nam Huh, and Sungyoung Lee. (2016) "Adaptive replication management in HDFS based on supervised learning." *IEEE Transactions on Knowledge and Data Engineering***28 (6)**: 1369-1382.

[19] Qu, Kaiyang, Luoming Meng, and Yang Yang. (2016) "A dynamic replica strategy based on Markov model for hadoop distributed file system (HDFS).", in *Cloud Computing and Intelligence Systems (CCIS), 2016 4th International Conference on*, IEEE : 337-342.

[20] S. Gopinath and E. Sherly, "A Dynamic Replica Factor Calculator for Weighted Dynamic Replication Management in Cloud Storage Systems", *Procedia Computer Science*, vol. 132, pp. 1771-1780, 2018.