IJCSE International Journal of Computer Sciences and Engineering Open Access
Review Paper Vol.-6, Issue-5, May 2018 E-ISSN: 2347-2693

A Review of Static and Dynamic Data Replication Mechanisms for Distributed Systems

R. Bhuvaneswari [1*], T.N. Ravi [2]

[1]Manonmaniam Sundaranar University, Tirunelveli
[1]Department of Computer Science, Periyar Govt. Arts College, Cuddalore, India
[2] Department of Computer Science, Periyar EVR College (Autonomous), Trichy, India

[*]*Corresponding Author: rbeswari17*@gmail.com

Available online at: www.ijcseonline.org

18/May/2018, Published: 31/May/2018

***Abstract*—**Huge dataset is widely used in various scientific applications. Hence, data replication is highly required to manage large volumes of data in a distributed manner. This improves the data access rate, reduces access latency and increases data availability. This paper presents a comprehensive review of the existing static and dynamic replication mechanisms along with the features. Static mechanisms determine the location of replication nodes during the design phase while the dynamic mechanisms select the replication nodes at the runtime. The dynamic replication approaches allow their associated replication strategies to be adjusted at the runtime according to the changes in user behavior and network topology. Also, they are applicable for a service-oriented environment where the number and location of the users who intend to access data often have to be determined in a highly dynamic fashion.

***Keywords*—** Cloud, Data Grid, Dynamic Replication, Static Replication

I. INTRODUCTION

Huge amount of data in the range of terabytes and petabytes is an important part of the shared resources in the scientific applications. Data replication is a strategy for creating multiple copies of data and stored them at multiple places [1, 2]. Data replication techniques are classified as static and dynamic replication techniques. In the static replication technique [3-7], the number of replicas and host node is predetermined. In the dynamic replication mechanism [8-16], the replicas are created and removed automatically based on the changes in the user access pattern, storage capacity and bandwidth. The dynamic replication mechanism makes intelligent choices regarding the data location based on the information of the current situation. But, it is difficult to collect the run time information of all the data nodes in a complex cloud infrastructure and hard to maintain consistency of data file [17]. The static and dynamic replication algorithms are further classified as distributed [3, 18-20] and centralized algorithms [9, 21-24].

In a distributed system, the main issue is to provide maximum data availability to the researchers. The size of the data to be accessed is in terabytes or petabytes. Efficient access of huge data is reduced due to the network latencies and bandwidth problems. With the increasing size of the data grid, there is an increase in the complexity of the architecture. High data availability is a main issue in the data grid environment. Hence, maintaining a local data copy is expensive. It is highly difficult to manage with high network latency, storage capacities at different sites and data availability. Data replication is a main approach to meet the challenges of high data availability. It promotes high data availability, fault tolerance, improved scalability, low bandwidth consumption and response time [25-29]. When data is replicated, multiple copies of data files are created and stored at different places in the grid or data center. Data replication can save storage resources as compared to the storage occupancy of data present at each site. For high-speed data access, data replication is an excellent tradeoff between storage availability and network bandwidth availability [30]. The main idea of the replication is to maintain the data proximate to the user for the efficient data access [31].

Replication techniques are classified as static and dynamic. In a static replication, the number of replicas and the host node is chosen statically at the initial stage. No more replicas are created or migrated after that [32, 33]. The dynamic strategies adapt to the changes in user requests, storage capacity and bandwidth and create or delete replicas on the new nodes depending upon the global information of the data grid [8, 25, 30, 34]. The dynamic strategies are better than

the static strategies as they can make intelligent decisions about the placement of data depending upon the information of the storage environment.

This paper presents a review of the static and dynamic data replication mechanisms. The main purpose of this review is to present the taxonomy of the existing replication techniques along with the significant advantages and issues. The main contributions of this review are stated as

- Providing the basic concepts and terminologies used in the field of data replication.
- Discussing about the two main types of data replication mechanisms: static and dynamic replication mechanisms
- Presenting the taxonomy of the mechanisms and highlighting their features.

The remaining sections in the paper are systematized in the following order: Section II describes the static replication mechanisms along with the features. The taxonomy and features of the dynamic replication strategies is presented in Section III. The performance metrics focused is described in Section IV. Section V comes up with the conclusion of the review.

II. STATIC REPLICATION

Ghemawat et al. [3] developed a Google File System (GFS) method for static data replication. Static replication mechanisms provide quick response, high availability and efficiency. But, it consumes more storage and energy resources. With the increase in the number of data replicas, there is an increase in the energy consumption.

Cibej et al. [6] explored static approach to data replication to replicate the files to make all sites as suitable for job executions. This approach achieves faster job scheduling, high fault-tolerance and no overhead due to dynamic data replication. A mathematical model of a grid is created and the optimization problem of static data replication on the grids is defined formally. The static optimization problem is NP-hard and non-approximable. Thus, solving the problem in a dynamic environment will be more difficult.

Loukopoulos and Ahmad [7] developed an object transfer cost model for large-scale distributed system. Adaptive Genetic Replication Algorithm (AGRA) is proposed for providing the replication scheme and the changes in the read and write requests for particular objects. The replica distribution is adapted quickly to reflect the new demands. The Genetic Algorithm (GA) constantly outperforms the greedy method in terms of running time and solution quality.

Long et al. [17] proposed a strategy for managing the replications in cloud storage cluster using the artificial immune system. Near optimal solutions are obtained by balancing the tradeoff among the mean file unavailability, service time, load variance, energy consumption and access latency. The proposed approach is energy effective and yields better load balancing for large-scale storage cluster. To minimize the energy cost, the number of data replicas should be as low as possible. GFS considers definite factors while making choices on the data replications. The new replicas are introduced on the chunk servers with least disk space consumption. The static replication mechanisms follow the deterministic policies. Hence, the number of data replicas and host nodes is predefined. Thus, it is simple to implement. The limitation of the static replication algorithm is the usage of a fixed replica number for all the files, as it does not provide best solution for data [17].

Khan and Ahmad [25] analyzed ten static replication techniques for fine-grained replication of frequently accessed data objects onto a set of selected sites to reduce the average access time perceived by the end users. A unified cost model that captures the minimization of the total object transfer cost is presented. This leads to the effective utilization of storage space, replica consistency, fault-tolerance and load-balancing.

Cidon et al. [26] proposed a MinCopysets method for re-randomizing the data replication to achieve better data durability properties. Randomized node selection is applied for data distribution. The proposed method can support any data locality or network topology requirements of the storage system. It does not present any overhead on the standard storage operations. The proposed method improves the data durability and reduces the disk bandwidth.

Qu and Xiong [27] introduced a high efficient method to yield high data availability while reducing the low replication cost. This algorithm can adapt the number of data replicas according to the traffic variations. The main advantage is minimum replication cost and replication failure possibility. However, the lookup path length and response time cannot be reduced significantly. This method achieved high replication rate, query efficiency, availability and reasonable path length at minimum cost.

Lee et al. [28] presented file assignment algorithms based on the open queuing networks for minimizing the load balance across all disks simultaneously and variation in the service time at each disk. An offline sort partitioning algorithm is presented for assigning similar access time to each disk to reduce the mean response time. An online hybrid partition algorithm approximates the behavior of sort partition and assigns groups of files with similar service times in

sequential intervals while assuring the load imbalance at any point does not exceed a certain threshold. The average response time of the proposed algorithm is minimum.

Hassan et al. [29] explored an optimization approach for replica management based on the objectives such as access latency, storage costs and data availability. Two novel algorithms are proposed for deciding the number of data replicas and placement of replica within the overlay. The computational efficiency is high and high quality solutions are obtained.

Zeng and Veeravalli [30] determined the number of Meta Data Servers (MDS) to achieve the minimum Mean Response Time (MRT) of all the metadata requests. A new metadata request balancing algorithm is proposed based on the request arrival rates, to find near-optimal solutions using a theoretical proof. The distributed strategies are devised to achieve minimum MRT of all requests arriving on MDS within cloud data centers.

Chen et al. [31] proposed a scalable replication solution for satisfying the consistency requirements in service-oriented and cloud-based applications. A Multi-fixed Sequencer Protocol (MSP) aims at assuring the satisfaction of the consistency models in a region and Region-based Election Protocol (REP) is responsible for flexibly balancing the workload among the sequencers by selecting new sequencers upon the presence of failure and distributes the loads to multiple sequencers. The impact caused by the crash failure is reduced.

Summary of static mechanisms

Static generation of maximum number of service replicas may guarantee the required performance at a high operation cost [32, 33]. The static replication strategy maintains the maximum number of active service replicas with a random policy [34]. However, static replication strategies are not often used as it does not adapt according to the environment [35]. Table I presents the comparison of the static replication mechanisms and Table II shows the features of static replication mechanisms.

TABLE I COMPARISON OF THE STATIC REPLICATION MECHANISMS

Replication Mechanisms	Performance Metrics	Advantages	Drawbacks
GFS [3]	• Read rate • Write rate • Append rate	• High data availability • Low response time • High reliability • Medium load balancing	• High energy consumption • High replication and storage cost • High bandwidth consumption
Cibej et al. [6]		• Faster job scheduling • High fault-tolerance • No overhead	• Solving the static optimization problem is difficult
AGRA [7]	• Network Transfer Cost (NTC) savings • Replicas • Execution time	• Quick adaptability to the changing environment • Improved solution quality • High speed • Minimum execution time • Low NTC	• The quality of solutions obtained is not higher than the static genetic algorithm method.
MORM [17]	• Power consumption • Total objective function • Mean file unavailability • Mean latency • Replication Factor • Mean service time • Load variance • Objective function value	• High data availability • High reliability • Lower storage cost • Lower energy consumption • Lower response time	• High bandwidth consumption
MinCopysets [26]	• Probability of data loss • Expected lost chunks	• High data durability • High availability • Low storage cost	• High energy and bandwidth consumption • High response time • Low reliability
RFH [27]	• Replica utilization rate • Replica number • Average replica number • Replication cost	• High data reliability • High fault-tolerant • High data availability Low bandwidth consumption	• High energy consumption • High response time

	• Average replication cost • Migration time • Average migration time • Migration cost • Average migration cost • Lookup path length • Load imbalance • Virtual node number	• Low replication cost	
Lee [28]	• Access rate and File size • Average response time	• Low replica utilization rate • Low replication cost • Low migration cost • Low migration time • Minimum load balance • Minimum lookup path length • High robustness	• Low data consistency
MOE [29]	• Reliability • Execution time	• High scalability • High performance • Low access latency • Low execution time	• High energy consumption • High replication cost • High bandwidth consumption
Optimal metadata [30]	• Probability of object requests • Mean response time • Processing rate	• High load balancing • Low delay of path • High scalability • Low response time	• High replication cost
SSOR [31]	• Time taken • Scalability • Overhead • Impact of lazy replication	• High scalability • High data availability • High load balancing	• High latency • High replication cost

TABLE II FEATURES OF STATIC REPLICATION MECHANISMS

Approaches	**Availability**	**Response time**	**Reliability**	**Bandwidth consumption**	**Load balancing**	**Access cost**	**Replication cost**	**Storage cost**	**Energy consumption**	**Migration cost**	**Consistency consideration**
AGRA	★★	★★	★★★	★★★	★★★	★★	★★★	★★★	★★	★★★	No
Cibej et al.	★★★	★	★★	★	★★★	★★★	★★	★★★	★	★	No
RFH	★★★	★	★★★	★★★	★★★	★★	★★★	★★★	★★	★★	No
GFS	★★★	★★★	★★★	★	★★★	★	★	★	★	★	No
MOE	★	★	★	★	★★★	★★★	★★	★★★	★	★	No
MinCopySet	★★★	★	★	★	★★★	★★★	★★	★★★	★	★	No
MORM	★★★	★★★	★★★	★	★★	★★	★★	★★★	★★★	★	No
Lee	★★★	★	★★★	★	★★	★	★★★	★	★	★★★	No
Optimal metadata	★★	★★★	★★	★★	★★★	★★	★	★	★★	★★	No
SSOR	★★★	★★	★★	★★	★★★	★	★	★	★★	★★	Yes

III. DYNAMIC REPLICATION

Lamehamedi et al. [8] developed a dynamic memory middleware that allows the Grid nodes for automatic data replication. The replication decisions are made based on the cost-estimation model and driven by the estimation of the data access gains and generation and maintenance costs of data replicas. A cost function that dynamically evaluates the replica placement policy is utilized by comparing the replica maintenance costs and data access gains of creating a data replica at any given location. Chang and Chang [9] presented a dynamic data replication mechanism for file replication in data grid. A popular file for replication is selected and a suitable number of replication copies and grid sites are computed. The importance of each record is distinguished by connecting a different weight to each historical data access record. The network load is reduced and a popular file is replicated to a suitable site.

Huang et al. [11] presented a new technique for the dynamic placement of data copies in the free blocks of file system according to the disk access patterns observed at the runtime. The adjacent replica that provides quick access can improve the performance of disk I/O operations, for accessing one or more replicas in addition to their original data block. As the layout of the file system is modified by using the free/unused disk space, the users are completely unaware of the modification. The energy consumption per data access is reduced due to the significant reduction in the disk access time.

Li et al. [13] presented a new dynamic replication strategy to reduce the storage cost and meet the data reliability requirements simultaneously. The number of data replicas in a data center and energy cost of the cloud storage system are reduced while satisfying the reliable requirements.

Saadat and Rahmani [14] proposed a new algorithm for dynamic data replication in data grids by using the prefetching technique. The future needs of grid sites are predicted according to the file access history. The files are pre-fetched to requester grid sites before receiving the requests. The overall system performance is increased by reducing the response time, access latency and bandwidth consumption.

Tang et al. [16] proposed a new architecture for supporting efficient data replication and job scheduling in the data grid environment. Two centralized and a distributed dynamic data replication algorithms are applied for improving the data access performance. The computing sites are organized into individual domains according to the network connection, and a replica server is placed in each domain. The dynamic replication can reduce the job turnaround time in a significant way. Wei et al. [20] developed a replica management scheme for large-scale cloud storage system. A new model is developed for acquiring the relationship between the data availability and number of data replicas. The workload is distributed dynamically among the data nodes by adjusting the number of replicas and node location according to the change in the workload and capacity of the data node.

Sun et al. [23] introduced a dynamic data replication strategy for distributed computing environment. The relationship between the system availability and number of data replicas is analyzed. A replication operation is activated when the popularity data permits a dynamic threshold. The appropriate number of data replicas is calculated and the replicas are placed among the data nodes in a balanced way. The efficiency of the cloud computing environment is improved.

Gill and Singh [35] proposed a dynamic re-replication and rebalancing strategy for heterogeneous cloud environment. The replication cost is optimized using the knapsack problem concept. The re-replication is performed if the data availability is greater than or equal to desired data availability. The replicas are placed at the high-cost data center or re-replicated at the low-cost data center, if the replication cost is higher than the user budget.

Boru et al. [36] developed a data replication strategy for the combined optimization of energy consumption and bandwidth capacity of the data centers in the cloud environment. The communication delay is optimized for providing high quality experience to the users. Tran et al. [37] introduced a data replication scheme (S-CLONE) for improving the efficiency of social network by considering the social relationship of data. S-CLONE resulted in better balancing of storage and writing loads across the servers. S-CLONE remains continuously superior to random replication irrespective of the changes in the social graph, number of deployed servers or the number of required data replicas.

Ranganathan and Foster [38] discussed the necessity for dynamic replication strategies to manage large data sets in a high performance data grid. The main advantage of the dynamic data replication is the automatic generation and removal of replicas according to the changes in the access patterns. The benefits of data replication are ensured even if the behavior of the user is changed frequently. The best replication strategies have significant savings in latency and bandwidth if the access patterns contain a small degree of geographical locality.

Mansouri et al. [39] proposed a dynamic data replication algorithm for improving the file access efficiency. Data replication should be used wisely due to the limited storage capacity of each grid site. The replicas are replaced based on the last time the data replica is requested, number of access and size of data replica. The best replica location is selected based on the data transfer time, storage access latency and distance between the nodes. The replication and scheduling

strategies yield better performance than the existing algorithms.

Bsoul et al. [40] devised a dynamic replication strategy to determine if the requested data replica should be copied to the node in the cluster by using a threshold. If the decision taken based on the threshold value is storing the new replica, then the new replica is considered significant than the group of replaced replicas. The proposed strategy achieved better performance in terms of total response time and total bandwidth consumption. Abad et al. [41] formulated a distributed adaptive data replication and placement algorithm that supports the scheduler to achieve better data locality. The job turnaround time is reduced by 16% in dedicated clusters and 19% in virtualized public clouds. The proposed algorithm does not incur any extra network usage.

Zhuo et al. [42] proposed a packet-level replication for the Delay-Tolerant Network (DTN). A centralized solution is provided for the better utilization of the limited storage buffers and contact opportunities. Data replication is performed based on the popularity and availability of the data item. The replication problem is formulated as a Mixed Integer Programming (MIP) problem. As the total storage space is not enough to fulfill the demand for all data items, some data items cannot obtain enough storage space. Hussein et al. [43] developed a data replication strategy for the adaptive selection of the data files that require replication to improve the cloud system availability. The proposed strategy decides dynamically the number of replicas and effective data nodes for replication. A new replica is created on a new block that achieves better new replication factor. The number of new data replicas is determined adaptively based on improving the file availability in a heuristic way.

Wang et al. [44] proposed a dynamic replication strategy based on the historical data access records and proactive deletion method. The number of data replicas is controlled to reach optimal balance between the read access time and the write update overhead. Different weights are applied to the access records at different time periods for finding a popular file. The popularity of a file is used for making decision about the replication of file. The proposed algorithm incurred lower data transfer cost than the existing algorithms.

Andronikou et al. [45] presented a set of algorithms for dynamic replication in a grid environment. The proposed solution can handle the dynamicity of the grid environment by increasing or reducing the set of data replicas based on the number and layout of the data requests. The replicas can be created or deleted automatically without incurring additional cost, execution time, network overhead, workload and network bandwidth. Sashi and Thanamani [46] proposed a modified Bandwidth Hierarchy Replication (BHR) algorithm for dynamic data replication in the grid computing environment. Unnecessary data replication is avoided by replicating files within the header region and storing the replicated files in the most frequently accessible sites. Thus, the storage space can be saved.

Lin et al. [47] proposed High-Quality of Service (QoS) First-Replication (HQFR) algorithm for data replication in large-scale cloud computing systems. The node combination techniques are applied for reducing the computational time of the QoS-Aware Data Replication (QADR) problem. The QADR problem is transformed to the Minimum-Cost Maximum-Flow (MCMF) problem to obtain the optimal solution. Thus, the total replication cost is reduced. Bai et al. [48] developed a strategy for managing the data replication in the cloud storage system based on the response time (RTRM). The replica creation is decided and best replica node for the users is selected based on the prediction of average response time. RTRM strategy involves three levels such as replica creation, selection, and placement. RTRM sets a threshold for the response time. A new data is replicated and the number of replicas is increased, if the response time is greater than the threshold. The bandwidth among the replica servers is predicted and the replica selection is performed accordingly while receiving the new request. The replica management strategies are improved in terms of network usage and service response time.

Gopinath and Sherly [49] proposed an effective data replication strategy based on the data access popularity. The data is categorized as hot, warm and cold based on the access patterns and replication of each category is managed. Minimum replica factor is set for the cold data. The replication factor of the hot and warm data is maintained according to its availability requirement. From the experimental results, it is proven that the proposed strategy is efficient and cost effective than the default replication management scheme in Hadoop Distributed File System (HDFS).

Azari et al. [50] introduced Popular Groups of Files Replication (PGFR) algorithm for the dynamic replication of data based on the assumption that the users in a Virtual Organization have similar interests in the groups of files. PGFR creates a connectivity graph to recognize a group of dependent files in each grid and replicates the most Popular file Groups of Files, thus increasing the local availability. PGFR algorithm reduced the mean job execution time, bandwidth consumption and avoided unnecessary data replication.

Amjad et al. [51] presented a survey about the classification of dynamic replication strategies in the data grid environment. From the survey, it is observed that there is no standard architecture for the data grid and a single strategy that addresses all issues related to the replication.

Tos et al. [52] conducted a survey of the recent dynamic replication strategies for the data grid environment. The key points of the strategies are discussed according to important metrics. Some important issues and open research problems are pointed out in the survey.

Replication is the process of copying and maintaining database objects such as tables in multiple databases to create a distributed database system. There are two types of replication called as eager or synchronous replication and lazy or asynchronous replication. Lazy replication is used for managing the agricultural loan database. This system can support reliable replicated data for agricultural loan system [53].

Cloud computing system has emerged as a key technology to enable access to the remote computing and storage infrastructures to the individual users and business enterprises. To achieve highly available and high performance services, the cloud data storage depends on the data replication. However, the replication technique brings the stability issues. The data is replicated in multiple geographically distributed data centers. To satisfy the increasing requirements of distributed applications, the cloud data storage adapts eventual stability and allows execution of the data intensive operations under low latency. Reliability is improved in the cloud computing environments because the CSPs utilize multiple redundant sites for disaster recovery. A novel Data Stability as a Service model is proposed to provide high stability in cloud using the crypto analysis algorithm [54].

Kathuria [55] presented a survey related to the security in the multi-cloud environment. Various approaches including file and application replication, partitioning of application system, application logic, application data, Byzantine protocol, Raincloud system and DepSky architecture are discussed to provide better cloud data storage decision to each customer.

Ganesan et al. [56] developed an algorithm for Byzantine Fault-Tolerant (BFT) replication using virtualization to deploy replication and service diversity in intrusion tolerant system. A prototype is developed to reduce the response time and increase the throughput.

Summary of dynamic mechanisms

Dynamic strategies automatically create and remove the replicas according to the variations in the data access pattern, data storage capacity and bandwidth. Most dynamic replica management strategies create a new data replica of the popular data based on the data access frequency. Thus, the replica creation always occurs at the end of each time interval [48]. Allowing updates on the replicas creates data consistency problems and introduces a significant amount of management overhead [52]. The comparison of the dynamic replication mechanisms is summarized in Table III. Table IV presents the features of dynamic replication mechanisms.

TABLE III COMPARISON OF THE DYNAMIC REPLICATION MECHANISMS

Replication Mechanisms	**Performance Metrics**	**Advantages**	**Drawbacks**
LALW [9]	• Total job execution time • Effective network usage • Storage resource usage	• Low network usage • Low memory consumption	• High job execution time
FS2 [11]	• Access time • Disk sector number • Average response time • Runtime • Energy improvement	• Low energy consumption • Minimum response time • Improved disk Input/Output I/O performance • Low replication overhead • Low seek time and rotational delay	• High energy consumption
CIR [13]	• Average number of data replicas • Storage cost ratio • Price	• High cost-effective • Low data storage cost • High data reliability	• Lower data reliability
PDDRA [14]	• Mean job time • Effective network usage • Total number of data replications • Hit ratio	• Low job execution Time • Effective network usage • Low number of data replications, • Better hit ratio	• Low scalability • Low fault tolerance

	• Percentage of storage filled	• High percentage of Storage Filled	
CDRM [20]	• Availability • Replica number • Average latency • System utilization rate	• High data availability • Low bandwidth consumption • Low data access cost • High load balancing	• Low reliability • High energy consumption • Low response time
D2RS [23]	• System bytes availability ratio • Replica number • Response time • Successful execution ratio	• High data availability • Successful execution rate • Low response time • Good convergence rate • Minimum number of replicas	• High user waiting time • Low data access • Low data availability
DCR2S [35]	• Replication cost • Average file probability • System byte effective rate	• Low replication cost • High reliability • High availability	• Low consistency rates • Low load balancing • High response time
Energy efficient data replication [36]	• Energy • Residual bandwidth • Data access delay • Bandwidth	• High data availability • Low energy consumption • Low response time • High Quality of Service • Minimum network congestion • High reliability	• The bandwidth usage increases to over 35 Gb/s, as data updates begin to propagate from Datacenter DB to multiple Rack DBs.
S-CLONE [37]	• Read cost • Improvement ratio • Coefficient of variation (CV) • Migration cost • After/before ratio	• Good balancing of storage and write loads across the servers • High replication efficiency • High load balance • High data availability • Low migration cost	• Low data availability
Fast spread [38]	• Percentage saving • Response time	• Low latency • Low bandwidth usage • Low response time	• High storage requirements
MDHRA [39]	• Job execution time • Number of inter-communication • Effective network usage	• Low response time • Low data transfer time • Low storage access latency • Effective network usage	• Low replica consistency
MFS [40]	• Total response time • Total bandwidth consumption	• Low response time • Better reduction in bandwidth consumption	• More bandwidth consumption
DARE [41]	• Proportion of node pairs • Number of file accesses • Fraction of files • Data locality • Geometric mean turnaround time • Mean slowdown • Data node locality • Average blocks created per job • Data locality of jobs • Coefficient of variation	• High data locality • Adequate file replications • Low network overhead	• High mean turnaround time and slowdown
DARA [42]	• Successful data retrieval probability • Number of packets in the buffer	• High successful data retrieval probability • High contribution gain	• The data retrieval probability is reduced due to the wastage of contact opportunities.
Adaptive replication [43]	• Average response time	• Low replication cost	• Low data access rate • Low data availability
CAGW_PD [44]	• Data transfer cost (DTC) ratio • Number of replicas	• Lower DTC • High storage capacity	• Low replication performance

QoS [45]	• Execution time ratio • Mean position	• Low execution time • Low replication cost	• Low Quality of Service
Modified BHR [46]	• Mean job execution time • Total number of replications • Total number of local file accesses • Total number of remote file accesses Percentage of storage filled/available • Probability of effective network usage	• High data availability • Low storage space consumption • Low mean job execution time • Efficient network usage	• Maximum network bandwidth is consumed due to the presence of replica in the site even if there is free storage.
HQFR [47]	• Total replication cost • Average recovery time • Worst recovery time • QoS violation rate • Execution time	• Low replication cost • High availability • High scalability	• High time complexity • High bandwidth consumption
RTRM [48]	• Average job time • Network utilization	• High performance • Low response time • High rapid data download • Low energy consumption • High data availability	• Low reliability • Low load balancing • High replication cost
WDRM [49]	• Access weight • Replication frequency • Storage space utilization • Mean storage space utilization • Average response time	• Minimum number of replications • Less storage space • Low response time	• The replication factor is set for the files manually
PGFR [50]	• Mean job execution time • Total number of replications • Effective network usage	• Low access latency • Minimum bandwidth consumption	• High replication cost

TABLE IV FEATURES OF DYNAMIC REPLICATION MECHANISMS

Approaches	**Availability**	**Response time**	**Reliability**	**Bandwidth consumption**	**Load balancing**	**Access cost**	**Replication cost**	**Storage cost**	**Energy consumption**	**Migration cost**	**Consistency consideration**
LALW [9]	★★★	★★	★★★	★★★	★★★	★★★	★★★	★★★	★★★	No	No
FS2 [11]	★★	★★★	★★	★★★	★★★	★★★	★★★	★★★	★★	No	No
CIR [13]	★★★	★	★★★	★★★	★	★★★	★★★	★★★	★★★	No	No
PDDRA [14]	★★★	★★★	★★★	★★★	★★	★★★	★★★	★★★	★★★	No	No
CDRM [20]	★★★	★	★	★★★	★★★	★★★	★★★	★★	★	No	Yes
D2RS [23]	★★★	★	★★★	★★★	★★★	★★★	★★★	★★★	★★★	No	No
DCR2S [35]	★★★	★	★★★	★★★	★	★★★	★★★	★★	★★	No	No
Energy efficient data replication [36]	★★★	★★	★★★	★★★	★★	★★	★	★	★★★	No	No
S-CLONE [37]	★★	★	★★★	★★★	★★★	★★★	★★★	★★	★★★	★★★	No
Fast spread [38]	★★	★★★	★★★	★★★	★★	★★	★★★	★★	★★	No	No
MDHRA [39]	★★	★★★	★★★	★★★	★★	★★	★★★	★★★	★★★	No	No
MFS [40]	★★	★★★	★★★	★★	★★	★★★	★★★	★★★	★★★	No	No
DARE [41]	★★	★★★	★★★	★★★	★★	★★	★★★	★★★	★★★	No	No

DARA [42]	★★	★★★	★★★	★★	★★	★★★	★★★	★★	★★★	No	No
Adaptive replication [43]	★★	★★	★★★	★★	★★	★★	★★★	★★	★★★	No	No
CAGW_PD [44]	★★★	★★	★★★	★★★	★★	★★★	★★★	★★★	★★★	No	No
QoS [45]	★★★	★★	★★★	★★	★★	★★★	★★★	★★★	★★★	No	No
Modified BHR [46]	★★★	★★★	★★	★★	★★★	★★★	★★★	★★★	★★★	No	No
HQFR [47]	★★★	★★	★★★	★	★★	★★	★★	★★★	★★	No	Yes
RTRM [48]	★★★	★★★	★	★★★	★	★	★	★	★★★	No	No
WDRM [49]	★★	★★★	★	★★★	★	★	★★★	★★★	★★★	No	No
PGFR [50]	★★★	★★	★★★	★★★	★★	★★	★	★	★★★	No	No

IV. PERFORMANCE METRICS FOCUSED

Data availability: All the replication strategies aim to provide maximum data availability in all distributed database environments and data grids.

Reliability: When data replication increases the availability, the reliability is improved. More the number of replicas, the chance of proper servicing to the user's request are high.

Scalability: The scalability is provided depending upon the architecture selected for the data grid. Different architectural models support different levels of scalability. Scalability is highly dependent on the architecture model than the replication algorithm.

Adaptability: The nature of the grid environment is very dynamic. Nodes enter and leave the data grid very frequently. The replication algorithm must be adaptive to provide support to all nodes in a grid at any given time.

Performance: The performance of the data grid environment increases with the increase in the availability of data [51].

Response Time: Response time is defined as the time taken to access the data from the servers [57]. The response time depends on server capabilities, server load, network path characteristics e.g. propagation delay and path load [58].

V. CONCLUSION

From this review, it is concluded that there are still a lot of works to be prepared in the field of data replication in the cloud computing environment. There has not been a standard architecture for data replication in the cloud environment. Most of the discussed techniques used a hierarchal architecture. The modifications of the hierarchal architectures to the real cloud environment are very interesting.

Some strategies consider improvement in the reliability, scalability, data access, load balance and fault tolerance and reduction in the user waiting time. Some approaches consider about the conservation of the network bandwidth. Hence, developing a complete method to consider the important parameters of data replication problem in a cloud environment is a very challenging task.

This paper reviewed the existing static and dynamic replication strategies. The data replication mechanisms are classified as static and dynamic. Static approaches determine the locations of replication nodes during the design phase while dynamic ones select replication nodes at the run time. Replication strategies are to be adjusted during the runtime according to the changes in the user behavior and network topology. Dynamic replication is more appropriate for a service-oriented environment where the number and location of the users who intend to access data have to be determined in a highly dynamic fashion. In the static replication strategy, the number of data replicas and their locations is initially set in advance. Instead, dynamic replication strategy dynamically creates and deletes the data replicas according to the changing environmental load conditions. Finally, it is very interesting to improve the replica selection process by involving the users in determining their preferences. Still, the discussed mechanisms are expanded and a new replication strategy that supports replica management in terms of replica creation, deletion and placement is proposed, to reduce both job execution time and network traffic.

REFERENCES

[1] S. Goel and R. Buyya, "Data replication strategies in wide-area distributed systems," in *Enterprise service computing: from concept to deployment*, ed: IGI Global, 2007, pp. 211-241.

[2] O. Wolfson, S. Jajodia, and Y. Huang, "An adaptive data replication algorithm," *ACM Transactions on Database Systems (TODS)*, vol. 22, pp. 255-314, 1997.

[3] S. Ghemawat, H. Gobioff, and S.-T. Leung, *The Google file system* vol. 37: ACM, 2003.

[4] R. M. Rahman, K. Barker, and R. Alhajj, "Replica placement design with static optimality and dynamic maintainability," in *Cluster Computing and the Grid, 2006. CCGRID 06. Sixth IEEE International Symposium on*, 2006, pp. 4 pp.-437.

[5] K. Shvachko, H. Kuang, S. Radia, and R. Chansler, "The hadoop distributed file system," in *Mass storage systems and technologies (MSST), 2010 IEEE 26th symposium on*, 2010, pp. 1-10.

[6] U. Čibej, B. Slivnik, and B. Robič, "The complexity of static data replication in data grids," *Parallel Computing*, vol. 31, pp. 900-912, 2005.

[7] T. Loukopoulos and I. Ahmad, "Static and adaptive distributed data replication using genetic algorithms," *Journal of Parallel and Distributed Computing*, vol. 64, pp. 1270-1285, 2004.

[8] H. Lamehamedi, Z. Shentu, B. Szymanski, and E. Deelman, "Simulation of dynamic data replication strategies in data grids," in *Parallel and Distributed Processing Symposium, 2003. Proceedings. International*, 2003, p. 10 pp.

[9] R.-S. Chang and H.-P. Chang, "A dynamic data replication strategy using access-weights in data grids," *The Journal of Supercomputing*, vol. 45, pp. 277-295, 2008.

[10] S. Acharya and S. B. Zdonik, "An efficient scheme for dynamic data replication," 1993.

[11] H. Huang, W. Hung, and K. G. Shin, "FS2: dynamic data replication in free disk space for improving disk performance and energy consumption," in *ACM SIGOPS Operating Systems Review*, 2005, pp. 263-276.

[12] S.-M. Park, J.-H. Kim, Y.-B. Ko, and W.-S. Yoon, "Dynamic data grid replication strategy based on Internet hierarchy," in *International Conference on Grid and Cooperative Computing*, 2003, pp. 838-846.

[13] W. Li, Y. Yang, and D. Yuan, "A novel cost-effective dynamic data replication strategy for reliability in cloud data centres," in *IEEE ninth international conference on Dependable, autonomic and secure computing (DASC)*, 2011, pp. 496-502.

[14] N. Saadat and A. M. Rahmani, "PDDRA: A new pre-fetching based dynamic data replication algorithm in data grids," *Future Generation Computer Systems*, vol. 28, pp. 666-681, 2012.

[15] X. Sun, J. Zheng, Q. Liu, and Y. Liu, "Dynamic data replication based on access cost in distributed systems," in *Fourth International Conference on Computer Sciences and Convergence Information Technology, 2009. ICCIT'09.* , 2009, pp. 829-834.

[16] M. Tang, B.-S. Lee, X. Tang, and C.-K. Yeo, "The impact of data replication on job scheduling performance in the Data Grid," *Future Generation Computer Systems*, vol. 22, pp. 254-268, 2006.

[17] S.-Q. Long, Y.-L. Zhao, and W. Chen, "MORM: A Multi-objective Optimized Replication Management strategy for cloud storage cluster," *Journal of Systems Architecture*, vol. 60, pp. 234-244, 2014.

[18] A. Doğan, "A study on performance of dynamic file replication algorithms for real-time file access in data grids," *Future Generation Computer Systems*, vol. 25, pp. 829-839, 2009.

[19] K. Shvachko, H. Kuang, S. Radia, and R. Chansler, "The hadoop distributed file system," in *IEEE 26th symposium on Mass storage systems and technologies (MSST)*, 2010, pp. 1-10.

[20] Q. Wei, B. Veeravalli, B. Gong, L. Zeng, and D. Feng, "CDRM: A cost-effective dynamic replication management scheme for cloud storage cluster," in *IEEE International Conference on Cluster Computing (CLUSTER)*, 2010, pp. 188-196.

[21] M. Lei, S. V. Vrbsky, and X. Hong, "An on-line replication strategy to increase availability in data grids," *Future Generation Computer Systems*, vol. 24, pp. 85-98, 2008.

[22] R. M. Rahman, K. Barker, and R. Alhajj, "Replica placement design with static optimality and dynamic maintainability," in *Sixth IEEE International Symposium on Cluster Computing and the Grid, 2006. CCGRID 06.*, 2006, pp. 4 pp.-437.

[23] D.-W. Sun, G.-R. Chang, S. Gao, L.-Z. Jin, and X.-W. Wang, "Modeling a dynamic data replication strategy to increase system availability in cloud computing environments," *Journal of computer science and technology*, vol. 27, pp. 256-272, 2012.

[24] D. Nukarapu, B. Tang, L. Wang, and S. Lu, "Data replication in data intensive scientific applications with performance guarantee," *IEEE Transactions on Parallel and Distributed Systems*, vol. 22, pp. 1299-1306, 2011.

[25] S. U. Khan and I. Ahmad, "Comparison and analysis of ten static heuristics-based Internet data replication techniques," *Journal of Parallel and Distributed Computing*, vol. 68, pp. 113-136, 2008.

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

[27] Y. Qu and N. Xiong, "RFH: A resilient, fault-tolerant and high-efficient replication algorithm for distributed cloud storage," in *Parallel Processing (ICPP), 2012 41st International Conference on*, 2012, pp. 520-529.

[28] L.-W. Lee, P. Scheuermann, and R. Vingralek, "File assignment in parallel I/O systems with minimal variance of service time," *IEEE Transactions on Computers*, vol. 49, pp. 127-140, 2000.

[29] O. A.-H. Hassan, L. Ramaswamy, J. Miller, K. Rasheed, and E. R. Canfield, "Replication in overlay networks: A multi-objective optimization approach," in *International Conference on Collaborative Computing: Networking, Applications and Worksharing*, 2008, pp. 512-528.

[30] Z. Zeng and B. Veeravalli, "Optimal metadata replications and request balancing strategy on cloud data centers," *Journal of Parallel and Distributed Computing*, vol. 74, pp. 2934-2940, 2014.

[31] T. Chen, R. Bahsoon, and A.-R. H. Tawil, "Scalable service-oriented replication with flexible consistency guarantee in the cloud," *Information Sciences*, vol. 264, pp. 349-370, 2014.

[32] Y. Chen, A. Das, W. Qin, A. Sivasubramaniam, Q. Wang, and N. Gautam, "Managing server energy and operational costs in hosting centers," in *ACM SIGMETRICS performance evaluation review*, 2005, pp. 303-314.

[33] M. Lin, A. Wierman, L. L. Andrew, and E. Thereska, "Dynamic right-sizing for power-proportional data centers," *IEEE/ACM Transactions on Networking*, vol. 21, pp. 1378-1391, 2013.

[34] M. Björkqvist, L. Y. Chen, and W. Binder, "Optimizing service replication in clouds," in *Proceedings of the Winter Simulation Conference*, 2011, pp. 3312-3322.

[35] N. K. Gill and S. Singh, "Dynamic cost-aware re-replication and rebalancing strategy in cloud system," in *Proceedings of the 3rd International Conference on Frontiers of Intelligent Computing: Theory and Applications (FICTA) 2014*, 2015, pp. 39-47.

[36] D. Boru, D. Kliazovich, F. Granelli, P. Bouvry, and A. Y. Zomaya, "Energy-efficient data replication in cloud computing datacenters," *Cluster computing*, vol. 18, pp. 385-402, 2015.

[37] D. A. Tran, K. Nguyen, and C. Pham, "S-CLONE: Socially-aware data replication for social networks," *Computer Networks*, vol. 56, pp. 2001-2013, 2012.

[38] K. Ranganathan and I. Foster, "Identifying dynamic replication strategies for a high-performance data grid," in *International Workshop on Grid Computing*, 2001, pp. 75-86.

[39] N. Mansouri, G. H. Dastghaibyfard, and E. Mansouri, "Combination of data replication and scheduling algorithm for improving data availability in Data Grids," *Journal of Network and Computer Applications*, vol. 36, pp. 711-722, 2013.

[40] M. Bsoul, A. Al-Khasawneh, Y. Kilani, and I. Obeidat, "A threshold-based dynamic data replication strategy," *The Journal of Supercomputing*, vol. 60, pp. 301-310, 2012.

[41] C. L. Abad, Y. Lu, and R. H. Campbell, "DARE: Adaptive data replication for efficient cluster scheduling," in *Cluster Computing (CLUSTER), 2011 IEEE International Conference on*, 2011, pp. 159-168.
[42] X. Zhuo, Q. Li, W. Gao, G. Cao, and Y. Dai, "Contact duration aware data replication in delay tolerant networks," in *Network Protocols (ICNP), 2011 19th IEEE International Conference on*, 2011, pp. 236-245.
[43] M.-K. Hussein and M.-H. Mousa, "A light-weight data replication for cloud data centers environment," *International Journal of Engineering and Innovative Technology*, vol. 1, pp. 169-175, 2012.
[44] Z. Wang, T. Li, N. Xiong, and Y. Pan, "A novel dynamic network data replication scheme based on historical access record and proactive deletion," *The Journal of Supercomputing*, vol. 62, pp. 227-250, 2012.
[45] V. Andronikou, K. Mamouras, K. Tserpes, D. Kyriazis, and T. Varvarigou, "Dynamic QoS-aware data replication in grid environments based on data "importance"," *Future Generation Computer Systems*, vol. 28, pp. 544-553, 2012.
[46] K. Sashi and A. S. Thanamani, "Dynamic replication in a data grid using a modified BHR region based algorithm," *Future Generation Computer Systems*, vol. 27, pp. 202-210, 2011.
[47] J.-W. Lin, C.-H. Chen, and J. M. Chang, "QoS-aware data replication for data-intensive applications in cloud computing systems," *IEEE Transactions on Cloud Computing*, vol. 1, pp. 101-115, 2013.
[48] X. Bai, H. Jin, X. Liao, X. Shi, and Z. Shao, "RTRM: a response time-based replica management strategy for cloud storage system," in *International Conference on Grid and Pervasive Computing*, 2013, pp. 124-133.
[49] S. Gopinath and E. Sherly, "A Weighted Dynamic Data Replication Management for Cloud Data Storage Systems," *International Journal of Applied Engineering Research*, vol. 12, pp. 15517-15524, 2017.
[50] L. Azari, A. M. Rahmani, H. A. Daniel, and N. N. Qader, "A data replication algorithm for groups of files in data grids," *Journal of Parallel and Distributed Computing*, vol. 113, pp. 115-126, 2018.
[51] T. Amjad, M. Sher, and A. Daud, "A survey of dynamic replication strategies for improving data availability in data grids," *Future Generation Computer Systems*, vol. 28, pp. 337-349, 2012.
[52] U. Tos, R. Mokadem, A. Hameurlain, T. Ayav, and S. Bora, "Dynamic replication strategies in data grid systems: a survey," *The Journal of Supercomputing*, vol. 71, pp. 4116-4140, 2015.
[53] Z. Sann and T. T. Soe, "Agricultural Loan System Using Data Replication Method," 2017.
[54] R. Reka and T. Parithimarkalaignan, "Recovering Data Stability Service for Preserving Rational Data in Cloud Environment," 2017.
[55] S. kathuria, "A Survey on Security Provided by Multi-Clouds in Cloud Computing," *International Journal of Scientific Research in Network Security and Communication* vol. 6, pp. 23-27, 2018.
[56] Ganesan.T, Tamizharasan.P, and S. G. Murugan.S, "A Shared Memory Technique for Windows Environment through Virtualization," *International Journal of Scientific Research in Network Security and Communication*, vol. 1, pp. 17-22, 2013.
[57] S. K. Yadav, G. Singh, and D. S. Yadav, "ANALYSIS OF A DATABASE REPLICATION ALGORITHM UNDER LOAD SHARING IN NETWORKS," *Journal of Engineering Science and Technology*, vol. 11, pp. 193-211, 2016.
[58] T. Loukopoulos, I. Ahmad, and D. Papadias, "An overview of data replication on the Internet," in *Parallel Architectures, Algorithms and Networks, 2002. I-SPAN'02. Proceedings. International Symposium on*, 2002, pp. 31-36.

Authors Profile

R. Bhuvaneswari is Assistant Professor in Department of Computer Science, Periyar Govt. Arts College, Cuddalore, Tamil Nadu, India. She has 21 years of teaching experience. She is currently pursuing Ph.D. She has presented 4 papers in various national/international seminars and conferences. Her area of interest includes Distributed Processing, Networking and Data Mining.

T. N. Ravi is Assistant Professor and Research Co-ordinator of PG and Research Department of Computer Science, Periyar E.V.R. College (Autonomous), Tiruchirappalli, Tamil Nadu, India. He has 27 years of teaching experience and 15 years of research experience. His area of interest includes Parallel Computing, Data Mining, Networking and Image Processing. He has guided 30 scholars for M.Phil. and Ph.D. He has published more than 37 research papers in reputed international/national journals.

