

Replication Strategy with Comprehensive Data Center Selection Method in Cloud Environments

M. A. Fazlina, Rohaya Latip*, Hamidah Ibrahim and Azizol Abdullah

Faculty of Computer Science and Information Technology, Universiti Putra Malaysia, Serdang, 43400, Selangor, Malaysia

*Corresponding Author: Rohaya Latip. Email: rohayalt@upm.edu.my

Received: 07 June 2021; Accepted: 12 July 2021

Abstract: As the amount of data continues to grow rapidly, the variety of data produced by applications is becoming more affluent than ever. Cloud computing is the best technology evolving today to provide multi-services for the mass and variety of data. The cloud computing features are capable of processing, managing, and storing all sorts of data. Although data is stored in many high-end nodes, either in the same data centers or across many data centers in cloud, performance issues are still inevitable. The cloud replication strategy is one of best solutions to address risk of performance degradation in the cloud environment. The real challenge here is developing the right data replication strategy with minimal data movement that guarantees efficient network usage, low fault tolerance, and minimal replication frequency. The key problem discussed in this research is inefficient network usage discovered during selecting a suitable data center to store replica copies induced by inadequate data center selection criteria. Hence, to mitigate the issue, we proposed Replication Strategy with a comprehensive Data Center Selection Method (RS-DCSM), which can determine the appropriate data center to place replicas by considering three key factors: Popularity, space availability, and centrality. The proposed RS-DCSM was simulated using CloudSim and the results proved that data movement between data centers is significantly reduced by 14% reduction in overall replication frequency and 20% decrement in network usage, which outperformed the current replication strategy, known as Dynamic Popularity aware Replication Strategy (DPRS) algorithm.

Keywords: Cloud computing; data replication; replica placement; data center merits; replication algorithm

1 Introduction

The worldwide shared mass data consists of a broad range of ambiguities of data types from various digital platforms [1–4]. As the data is high in volume, it demands high storage to keep all the data safe. Therefore, cloud computing is the best choice in the current state of the art to facilitate mass space to store bulk data [5,6]. Cloud providers are vibrant, resilient and the most favored for users



This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

across the globe, as they offer multiple services, including Platform as a Service (PaaS), Software as a Service (SaaS), Quality as a Service (QaaS), and Infrastructure as a Service (IaaS) [7–11].

Cloud computing is not exempted from facing problems in providing consumers with a high availability data service without data sensitivity disadvantages as a secure multiple service provider [12–14]. As a result, a data management strategy is required to offer high data availability and efficient access for every user. Dynamic data replication, which stores several replicas at different data centers to improve system load balancing, is a promising solution for addressing this issue [15–18].

Data replication in the cloud environment is described as making multiple physical copies for each logical data object and locating replica copies in different locations or storage nodes [18–20]. Depending on the cloud replication goals, there are many ways to implement data replication in a cloud replication system environment. The respective goals have their disadvantages, which often degrade performance [21,22]. Finding the best data center to keep safe replicas is crucial for the replication process since it calculates essential variables when determining the best data center to store replica copies. Several strategies devised by researchers have been established to ensure that a good location for replica copies is determined. Hence, the most conversed issues among existing research work as a tough challenge in cloud replication environment are not limited to ineffective network usage, high replication frequency, high fault tolerance, extensive storage consumptions, and many more. Therefore, to overcome such performance issues, an established and systematic replication strategy must be created. Cloud providers will be able to provide enhanced performance to consumers with more significant data availability, quicker response time, low fault tolerance, decreased storage consumption and effective network usage with the requisite replication strategy [19,23,24]. The main contributions of this research work are:

- a. To study the current data center selection methods and identifies the research gaps for cloud replication environments.
- b. To propose Replication Strategy with Data Center Selection Method (RS-DCSM) to resolve inefficient network usage and minimize replication frequencies while identifying suitable data centers to replica copies.

The remainder of this paper has been structured as follows: Section 2 discusses the related works on replication strategies and data center selections in the cloud environment. Section 3 presents a detailed explanation of the proposed model, system architecture, parameters and configurations. Section 4 offers results and discussions of the experiments. Finally, Section 5 concludes the work and presents the future directions.

2 Related Works

Globally, data replication in the environment is evolving as an explicit data management technique in the cloud environment [25]. In data replication environments, there are two (2) common mechanisms for replication strategies. First of all, static replication is a predefined strategy for particular replica environments and is very easy to implement, but this strategy typically does not adapt to every environment [26,27]. The second mechanism is dynamic replication, known as agile replication strategies, where the algorithm can efficiently create and remove any replicas depending on the access trends of system users [19,28].

2.1 Static Replication

The static replication mechanism is recognized as a simple structure, yet often unfavored and not suitable to be adapted in complex cloud replication systems. However, despite the disadvantages of static replication approaches, many researchers have still accomplished their works by adapting static replication strategies [26]. The Multi-Objective Replication Management (MORM) algorithm was proposed to achieve multiple research objectives such as latency, data availability, service time, energy-saving for data centers and load balancing in [29] development work. The weakness discovered in MORM is when files arrive in batch patterns to be placed in storages. The algorithm must calculate and decide the needs of new replica placements based on previously allocated files, but the algorithm capability was limited by the static replication mechanism implemented in the architecture. Therefore, due to the static method limitations, this study has disadvantages of low data accessibility, high execution time, high replication cost and low reliability because it does not dynamically assign replicas based on current device needs.

Another research-adapted static replication mechanism is the MinCopySet algorithm. A fixed number of replicas are determined to achieve their target for faster response time and high data durability in this study. This strategy has practically improvised data resilience and reduced network latency. The limitation found in this algorithm is the over-use of such replicas due to replica placement in the same storage nodes, resulting in high energy consumption and poor data reliability [30].

Similarly, [31] employed a static approach to achieving load balancing between fixed replicas in their proposed Google File System (GFS) algorithms. The researcher could minimize the response time, but there were some drawbacks to the replica placement process in their approaches. By creating specific replicas for all files and placed at appropriate locations, the researcher attained the study goals. Subsequently, as the number of replicas for all files is pre-determined, regardless of the user access pattern, the implications faced by this research work are increases in energy consumption and high storage consumption.

2.2 Dynamic Replication

Many researchers and practitioners in different cloud environments such as; grid, cloud, edge and fog computing environments have widely adopted the dynamic replication mechanism due to its ability to handle data replication intelligently flexibly based on system users accessing patterns [32–34].

Relatively, [33] researchers have developed a popularity-aware multi-failure resilient and cost-effective replication (PMCR) algorithm with an identical strategy as PRCR to store replica copies into primary and backup tiers by splitting cloud storage. The goal is to increase data resilience in cloud storage, allowing the PMCR algorithm to distinguish hot, warm and cold data based on its popularity. The goal was accomplished in the study, yet the researcher has to accept process overheads that indirectly affect the response time due to the algorithm's multiple splitting activities.

Recently, researchers [24] proposed a dynamic replication algorithm, namely Hierarchical Data Replication Strategy (HDRS). Based on the prediction of subsequent access statistics for data files in the cloud, HDRS may detect popular files and replicate the replicas to the optimum location utilizing network-level locality. HDRS triggers the placement approach; otherwise, the replacement technique is used for storage clearance. According to the researchers, the HDRS successfully lowered response time, bandwidth, and latency. However, one of the study's flaws is the long replication time. The placement strategy's replication process overheads were influenced by a multi-hierarchy verification process, which the researcher ignored.

Researcher [35] focused their study on addressing the reliability issues of data storing by cloud providers using the dynamic replication approach. This study proposed integrated Location-Aware Storage Technique (LAST) into the open-source Hadoop Distributed File System (HDFS) called LAST-HDFS algorithm. The algorithm works as a monitoring manager, detects illegal data transfers in the cloud, and enables storage location of file moved during migration and replication process in a cloud environment. The research was successfully attained high security and privacy on the placement of migrated and replicated data in clouds. On the other hand, the disadvantages of this research lead to increased costs due to sophisticated security features. Additionally, this study also suffers from high network usage because the location monitoring and detection functions require data collection in real-time based.

2.3 Data Replications with Data Center Selection Methods

Data replications consist of many sub-strategies, techniques, methods, and algorithms that are coherently supported to establish comprehensive cloud replication strategies. Generally, there are 3 main phases under data replication: identifying popular data, determining the number of replicas, and placing replica copies. Numerous researchers have done great work to establish various algorithms to fulfil the requirement for respective data replication phases [36].

Researchers often incorporate data center selection approach into the data placement process in almost every replication strategy. In fact, the method is a distinct and huge part of the replication process, whereby critical factors are decided when selecting suitable data centers to store replica copies. Usually, proposed factors or parameters directly affect performance enhancement, and most of them focus on decreasing network usage and replication frequencies in cloud replication environments [37].

In 2016, Mansouri proposed Adaptive Data Replication Strategy (ADRS) in a cloud environment. ADRS deployed a data center selection criteria method by considering five (5) significant parameters; storage usage, load variance, latency, mean service time and failure probability. The cost function was calculated using stated parameters to retrieve fitness values for every data center known as sites in this research work. Reference [38] designed ADRS to choose the lowest cost function to be selected data center to store newly generated replicas. ADRS improvised a few performances, which are hit ratio and network usage. However, replication time is not considered in their measurement, impacted by the tedious computation and replication process completions.

Dynamic Popularity aware Replication Strategy (DPRS) proposed by [39]. The frequency of file requests, storage availability, and data center distances are used in their algorithm to pick the optimal data center. The weightage idea is used to compute merit in data centers, where system administrator interaction is required to define necessary weights based on system goals. With the parallel download idea and the proposed data center selection method, DPRS achieved efficient network consumption and reduced replication frequency. On the other hand, the researcher ignored fault tolerance, which could have been caused by inaccessible sites due to the elevated traffic. The system will therefore suffer from data loss as well as a long response time.

Researcher [15] achieved a similar aim through developing a systematic algorithm called Cost Function based on the Analytical Hierarchy Method for Data Replication Strategy (CF-AHP). In order to decide the best data center candidates to position newly created replicas, CF-AHP as multi-criteria optimization model was adapted to reduces energy consumption in data centers. The data center selection criterion consists of; mean service time, access rate, latency, load variance and storage usage. Despite achieving its goals, the researcher is unaware of the effect on the central database that during the replication process experiences a high update rate.

Researchers [35] proposed DMDR in their research, which ingested data center selection criterion to select the best data center in the cloud replication environment. This study enhanced storage utilization through introducing two (2) criteria in data center selection method; most central and number of accesses. This algorithm considers centrality to minimize data retrieval time, which will pick the most central data center as the best data center. In DMDR, an accumulation using proximity formulation was adapted (Newman, 2009), so the lowest value of distance summation will be selected as the most centralized data center. Additionally, computation on a greater number of access is counted to find a data center with the highest demand for a candidate file. Researchers sought to reduce the use of the network during file retrieval by adding this data center criterion. Conversely, the proposed criterion is not sufficiently faultless, resulting in system performance deterioration caused by high replication time.

Unlike other researchers, [40] proposed different ideology to place replica in data centers. Rather than using multiple data selection factors to determine the best data center, the researcher adapted static data placement paradigm to fit user access frequency patterns in social media and identifies an appropriate data center to place replica. The researcher emphasized that data placement as a dynamic problem solution and suggest an approach in social networks such as Facebook to address optimized data placement with tolerable latency and incurring minimal service costs. In the resolution, user access data are collected according to friends' connections and duration of communications. A replica access table is generated to record the frequency and every data center according to connections occurrences. The nearest data center for individual friends is identified to place the data to ensure latencies and replica creations are reduced concurrently. Thus, the researcher attained optimized data placement and reduced the monetary need to maintain the cloud's replication environment. However, the drawbacks yet found at high replication time to replicate data into storages due to data travels in long networks to verify replica placement requirements.

Researcher [41] recently developed a dynamic replication technique for addressing massive data movement around cloud data centers. The author suggested BDS+, a Bandwidth Dynamic Separation method for inter-data center data replication. The method attempts to improve data transfer performance by adjusting dynamic bandwidth separation, ensuring bandwidth allocation for online traffic by calculating traffic demand, and rescheduling bulk-data transfers for offline data services. It employs centralized architecture and application-level multicast on the network, with the central controller managing intermediate server data transmission. The study does not employ any specific selection methods to find the optimal data center, but it does appoint a manager to shift replicas to the proper storage using online and offline scheduling. The researcher successfully reduced bandwidth use, however, he failed to account for the long replication time. The technique takes longer to sort the traffic schedule than it does to start the replication process.

Recent research has introduced a bio-based Multi-Objective Particle Swarm Optimization (MO-PSO) and Ant Colony Optimization (MO-ACO). The researcher developed a novel intelligent approach for dynamic data replication in a cloud environment [27]. The first MO-PSO to select replica depends on the most requested by users. At the same time, MO-ACO was used to decide the best data center to store replica copies through comparing individual data centers based on shortest distance, data center with high access, storage capacity, output, and data center with a large number of hosts and virtual machines. The study achieved better replication costs by accelerating the response time and replication time also succeeded in enhancing network usage efficiency. However, the drawbacks overlooked by the researcher is the bio-based algorithms caused process time overheads and high replication frequencies.

The data center selection methods are holistically crucial to the cloud replication process. The methods developed have similar objectives to determine the best data center before replicas are stored. In order to ensure effective network usage and low replication frequency are achieved that eventually increase overall replication, a precise method with essential factors should be considered, which ultimately enhances overall replication performance in cloud replication environments. The detailed summary for each study in this subsection is shared in [Tab. 1](#).

Table 1: Summary of related works

Article	Algorithm	Advantages	Limitations
[15]	Analytical Hierarchy Method for Data Replication Strategy (CF-AHP)	<ul style="list-style-type: none"> • Low energy consumption 	<ul style="list-style-type: none"> • High response time • High update rate
[27]	Multi-Objective Particle Swarm Optimization (MO-PSO) and Ant Colony Optimization (MO-ACO)	<ul style="list-style-type: none"> • Efficient network usage • Low response time • Low replication time 	<ul style="list-style-type: none"> • High replication frequency • High process time
[35]	Data Mining-based Data Replication (DMDR)	<ul style="list-style-type: none"> • Efficient network usage 	<ul style="list-style-type: none"> • High replication frequency
[38]	Adaptive Data Replication Strategy (ADRS)	<ul style="list-style-type: none"> • High hit ratio • Efficient network usage 	<ul style="list-style-type: none"> • High replication time • Low data availability
[39]	Dynamic Popularity aware Replication Strategy (DPRS)	<ul style="list-style-type: none"> • Efficient network usage • Low replication frequency 	<ul style="list-style-type: none"> • Low fault tolerance • High traffic queue
[40]	Cost-effective dynamic data placement for efficient access of social networks	<ul style="list-style-type: none"> • Low latency • Low storage usage 	<ul style="list-style-type: none"> • High network usage
[41]	Bandwidth Dynamic Separation method for inter-data center data replication (BDS+)	<ul style="list-style-type: none"> • Efficient network usage 	<ul style="list-style-type: none"> • High process time • High replication frequency

3 Replication Strategy with Data Center Selection Method (RS-DCSM)

A non-comprehensive replica positioning method would result in an access skew whereby some of the data centers are heavily utilized, but some are idle. This scenario can lead to network congestions and cloud storage inconsistencies, leading to other performance degradation. The contributing factors for the high network consumptions in cloud environments are usually due to inefficient replication strategy, which can be consequence explicitly by inadequate data center selection method to place new replicas [24,39,42,43]. On the other hand, when replicas are successfully placed in appropriate data centers, efficient network usage, high fault tolerance, high data availability, and low replication frequency are achievable, ultimately providing better performance in a cloud replication environment.

Essential factors are crucial for determining and considering when new replicas are ready to be saved in storage. Consequently, the best-defined data center and replica allocation will result in minimal data movements. The reduced movements are because during replica copies are required for

rapid file recovery, the data center selection method provides faster replica accessibility for downloads in the most appropriate data center. Similarly, in the proposed RS-DCSM, we considered several substantial factors before choosing the appropriate data center to place the replica copies in storage nodes.

3.1 System Architecture

The overall system architecture was created the same way as the other work by [39]. We selected [39] to compare the competence of our proposed RS-DCSM because it achieved various goals and improved numerous performance measures in cloud replication, including reducing network usage and minimize replication frequency. Technically, [39] used architecture as Fig. 1 where clusters, data centers, Global Replica Manager (GRM), and a Local Replica Manager (LRM) are part of the system architecture. The GRM is the broker of the system, located in the cloud’s center and connected to other nodes by several routers and connections. The experiment architecture comprises multiple clusters interconnected to individual storage by a few data centers.

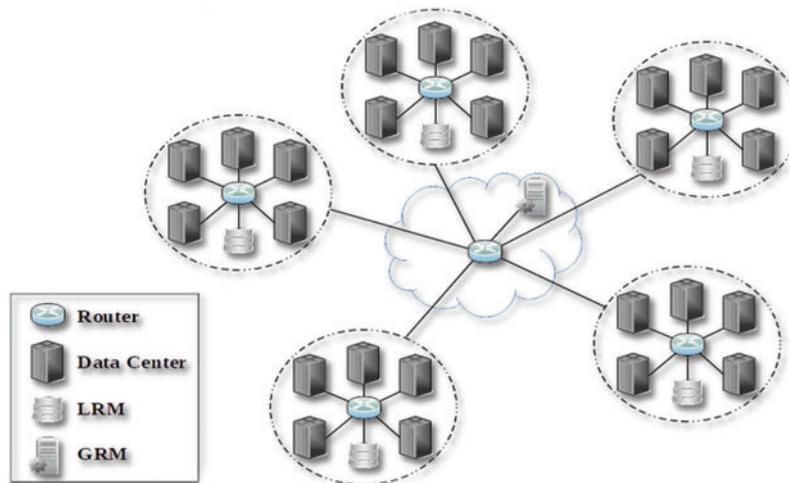


Figure 1: System Architecture

The specification of every node in Fig. 1 is summarized in Tab. 2. The simulation environment in this research work was configured using CloudSim and the parameters used to establish the simulation environment presented as in Tab. 2.

Table 2: Parameters

Parameters	Values
Total number of clusters	10
Total number of nodes	100
Numbers of nodes within the same clusters	10
Number of different files	200
Size of each file	From 1 to 20 (GB)
Storage size for every cluster nodes	60 (GB)
Number of files accessed by a job	3–10

(Continued)

Table 2: Continued

Parameters	Values
Round length	100
Number of intermediate nodes between two nodes in the same cluster	1
Number of intermediate nodes between two successive cluster	3
Inter-Router bandwidth	10 (Gbps)
Router-to-Site bandwidth	2.5 (Gbps)
User-to-Router bandwidth	100 (Gbps)
GRM-to-Router bandwidth	2.5 (Gbps)
LRM-to-Router bandwidth	1 (Gbps)
The Duration of round (Td)	1000 (s)
W1, W2, W3	1/3

3.2 Data Center Selection

Every standard replication environment has a central manager to manage the entire replication architecture. As for this research work, the manager is known as GRM, as in Fig. 1. In this research work, we assume candidate files for replication are ready in a selection list and recognized as *Most Popular File*, (MPF_i). Therefore, GRM is responsible for receiving the list of MPF_i from LRM.

Subsequently, the GRM, as a central unit in this system architecture, responsible for identifying MPF_i for individual clusters, C_j where j is cluster index $j = \{1, 2, \dots, n\}$. GRM will proceed to verify the existence of MPF_i in the requesting cluster. After GRM verifies the MPF_i is not existing in the requesting Cluster (C_j), GRM will send the replication file to the desired storage node.

Prior to that replication process, the RS-DCSM algorithm is initialized to select the most appropriate data center to place replica copies in storage. Therefore, RS-DCSM starts to identify data center merits (μ) which, is known as selection criteria for each data center (DC_x) x is data center index where $x = \{1, 2, \dots, w\}$ in C_j .

Three (3) factors must be computed to derive the primary equation for the RS-DCSM algorithm; *User Merit*, *Storage Merit*, and *Centrality Merit*. The individual factors recognized as selections criteria that acquire separate functions to calculate the merits values for every DC_x in the requesting C_j . The calculation is to ensure an accurate data center merit, μ is identified to select the best DC_x to place each MPF_i replica. The best or appropriate data center is identified to have the highest value of μ . The criteria of data center selection are, *User Merit* (μ), *Storage Merit* (σ) and *Centrality Merit* (λ). All three (3) criteria values are diverse in scale; hence, it is necessary to normalize their values into a scale between 0–1. Eventually, the final values of the μ is attainable. In this research architecture, the RS-DCSM algorithm resides in GRM, and the algorithm's main process is handled between GRM and LRM.

3.2.1 Selection Criteria

The discussion on the proposed criteria and the calculation for individual factors in merit values are as follows:

a. Accumulation of User Merit, (μ)

The μ is calculated based on a total file accessed ($\sum \hat{F}$) in each DC_x regardless of file name or Id. The greater number of files (F_i) accessed or requested in DC_x resulting in a higher value for $\sum \hat{F}$ which, is indicating the data center is popular. Therefore, chances of the same data center will be accessed near future for MPF_i downloads are highly possible. Authors of [24] and [44] stated the cruciality of considering geographical locality in a replication environment whereby when a file was accessed recently in a particular storage node, local nearby data centers have a high possibility of being re-accessed. The researchers undoubtedly agreed that placing replica copies in the data center with a high frequency of specific files (popular candidate file) is ineffective, instead, it is more recommended to place popular files in the popular data center.

Therefore, knowing the advantage, we proposed placing a file, MPF_i at the active data center with high user access rate. Therefore, a cumulative calculation on file access time is necessary to identify the best site with the highest value of *User Merit*, μ . Hence, in this research work, RS-DCSM is designed to choose data center with the greatest number of $\sum \hat{F}$ as one of the criteria to place replica copies using Eq. (1);

$$\mu = \sum \hat{F} \text{ in Data Center, } DC_x \quad (1)$$

In Eq. (1), μ is the total file access ($\sum \hat{F}$) for individual DC_x where x is data center index; $x = \{1, 2, \dots, w\}$. In order to retrieve accurate values, μ is calculated in a separated function for every DC_x , in C_j .

b. Accumulation of Storage Merit, (σ)

Availability of more space in storage gives higher opportunities for the data center to be chosen as the best candidate for replica storage [45–51]. Consequently, *Storage merit* accumulation in this research is to identify available storage space in each DC_x . The available storage in individual data centers, is computed using Eq. (2).

$$\sigma = \frac{\sum \text{AvailableSpace}}{\sum \text{Space}} \text{ in } DC_x \quad (2)$$

$\sum \text{AvailableSpace}$ denotes total free storage space, divided by $\sum \text{Space}$ referring to the total storage space allocated in every data center, DC_x at cluster, C_j .

c. Accumulation of Centrality Merit, (λ)

Centrality Merit, (λ) is accumulated through the summation of another two (2) sub-criteria.

i. Closeness Centrality, (CC)

As the first sub-criteria, this RS-DCSM identifies data center which has the shortest average distance from one data center, DC_x to other data centers in the same requesting C_j . This criterion is commonly used in choosing the best data center in almost every replication strategies known as *CC* [52,53]. The *CC* is computed by RS-DCSM using Eq. (3.1)

$$CC(x) = 1 \left(\frac{\sum d(x, y)}{\sum \text{diss}_j} \right) \quad (3.1)$$

$\sum d(x, y)$ denotes the total distance from one data center (x), to another data center (y), divided by $\sum \text{diss}_j$ referring to total distances of all data center distances in the same C_j . In Eq. (3.1), the data center centrality is obtained from the complement of summation of distance values.

ii. Degree of Centrality, (DCen)

The second sub-criteria is $DCen$ referring to a data center with a high number of connectivity or alternative network path to other data centers in the same requesting C_j [54]. $DCen$ is very practical to be adopted in selecting appropriate storage node to place replica copies. It is because in order to address any fault tolerance issues in the cluster environment $DCen$, capable to select another network route to retrieve replica copies during having any traffic bottleneck or server interruption issues in particular data center [54,55]. Adopting $DCen$ provides greater advantage to access information and the reliability even better compared to those data centers that have fewer connections [54,55]. Therefore, the $DCen$, in this research work is computed by the RS-DCSM algorithm using Eq. (3.2).

$$DCen(x) = \sum NumOfConnections \quad (3.2)$$

In Eq. (3.2), degree of centrality for a data center (x), $DCen(x)$ is directly obtained through calculating the total number of connections, $\sum NumOfConnections$ available for every data center (x).

Considering all the benefits gained through integrating both sub-criteria, λ is calculated using Eq. (4).

$$\lambda = CC(x) + DCen(x) \quad (4)$$

Finally, three (3) main merits criteria were explained in previous paragraphs. The third criteria, which consist of another two (2) sub-criteria, are described in detail. Therefore, eventually, $DataCenter Merit$; μ is derived, and the RS-DCSM algorithm is obligated to this primary equation as Eq. (5). μ obtained for respective data centers are normalized to scale between 0–1.

$$\mu = \mu + \sigma + \lambda \quad (5)$$

3.2.2 Determination of Data Center

The replication process will proceed after RS-DCSM identifies merit values as in Eq. (5) for individual data centers (x) in the cluster C_j . The obtained μ values are further sorted in descending order by LRM subsequently stored as $Seta_j$ and the $Seta_j$ passed to GRM. The GRM will choose N best data centers with the greatest values from $Seta_j$ and save them as $Setb_j$. The number N refers to the number of data centers that allow parallel downloads for MPF_i . Subsequently, GRM will segment one MPF_i into N data centers using Eq. (6). File fragmentation is calculated using Eq. (6), determining how much a file size can be chunked before delivering to N data centers.

A system administrator determines the N based on their system requirements. On a single MPF_i a higher number of N will result in more segments. The calculated values will be organized in descending order and enlisted in $Setb_j$ for the selection of the best data centers. The $Setb_j$ is made up of elements that will be used to admit segments from the MPF_i file. The file fragmentation formula adopted from [39] and calculated as Eq. (6).

$$Fragmentation, N_i[t] = \frac{Set_b[t]}{\sum_{t=1}^N Set_b[t]} \quad (6)$$

where $N_i[t]$ denotes list that includes the fragmentation percentages of MPF_i to be distributed into N data centers and t represents the item index in the $Setb_j$. Therefore, respective fragmented file, MPF_i will be sent for replication process and stored in the appropriate data center to enable parallel downloads.

Instead of randomly examining fundamental factors in choosing an acceptable data center to store the replica, the criteria in the RS-DCSM algorithm are meticulously calculated from multiple significant perspectives. Despite this, the RS-DCSM algorithm is successful at reducing network utilization while maintaining replication frequency. The improved replication performance is due to its ability to dynamically locate replicas in the most strategic location without sacrificing the ability to choose the best data center using the proposed RS-DCSM's proposed multi-criteria. Therefore, the proposed data center selection method (RS-DCSM) with all three (3) criteria; *User Merit*, (μ) *Storage Merit*, (σ) and *Centrality Merit*, (λ) is presented in Algorithm 1 and the RS-DCSM Flowchart is shared in Fig. 2 for better understanding on the process.

Algorithm 1: RS-DCSM

```

1. //List that contains the File Access
2.  $C_j \ni \{\hat{F}DC_1, \hat{F}DC_2, \dots, \hat{F}DC_x\}$  //  $\hat{F}$  = Total File Access in  $DC_x$ 
3. INPUT: Selected  $Cluster C_j$ ; //At GRM
4. For All Most Popular File,  $MPF_i$ ; { //,  $MPF_i$  is list of candidate file for replication
5.   For every Data Center,  $DC_x$  in Cluster,  $C_j$  Do; { //At LRM
6.     Apply Data Center Selection Method (RS – DCSM);
7.     //Function UserMerit,  $\mu$ 
8.      $f(\mu) = \text{Total File Access, } \sum \hat{F}$ , in DataCenter,  $DC_x$ ;
9.     //Function StorageMerit,  $\sigma$ 
10.     $f(\sigma) = \frac{\sum \text{AvailableSpace}}{\sum \text{Space}}$  in DataCenter,  $DC_x$ ;
11.    //Function CentralityMerit,  $\lambda$ 
12.     $f(\lambda) = \text{ClosenessCentrality, } CC(x) + \text{DegreeCentrality, } DCen(x)$ ;
13.    //LRM Calculate DataCenter Merit,  $\mu$ ;
14.    DataCenter Merit,  $\mu = \mu + \sigma + \lambda$ ;
15.    Sort DataCenter Merit,  $\mu$  in Descending Order;
16.    Store sorted  $\mu$  as  $Set_a$ ; } //  $Set_a$  is return to GRM
17.    Select  $N$  best DC from  $Set_a$ ; //Best DC = Higher DC Merit value
18.    Store  $N$  DCs as  $Set_b$ ;
19.    Replica Fragmentation,  $N_i[t] = \frac{Set_b[t]}{\sum_{i=1}^N Set_b[t]}$  for  $Set_b$ ;
20.    // Fragmentized  $MPF_i$  sent to  $N$  DCs for replication
21.    //  $MPF_i$  replication completed
22.    End.}

```

4 Results and Discussions

The capability of RS-DCSM is proved through conducting few experiments to measure improvements in cloud replication performances. Similar to the benchmark study by [39], this research work selects two (2) best data centers; N is fixed to 2, where the selected data centers are listed in Set_b . There are two (2), performance metrics was measured to perceive the enhancements of proposed RS-DCSM as below.

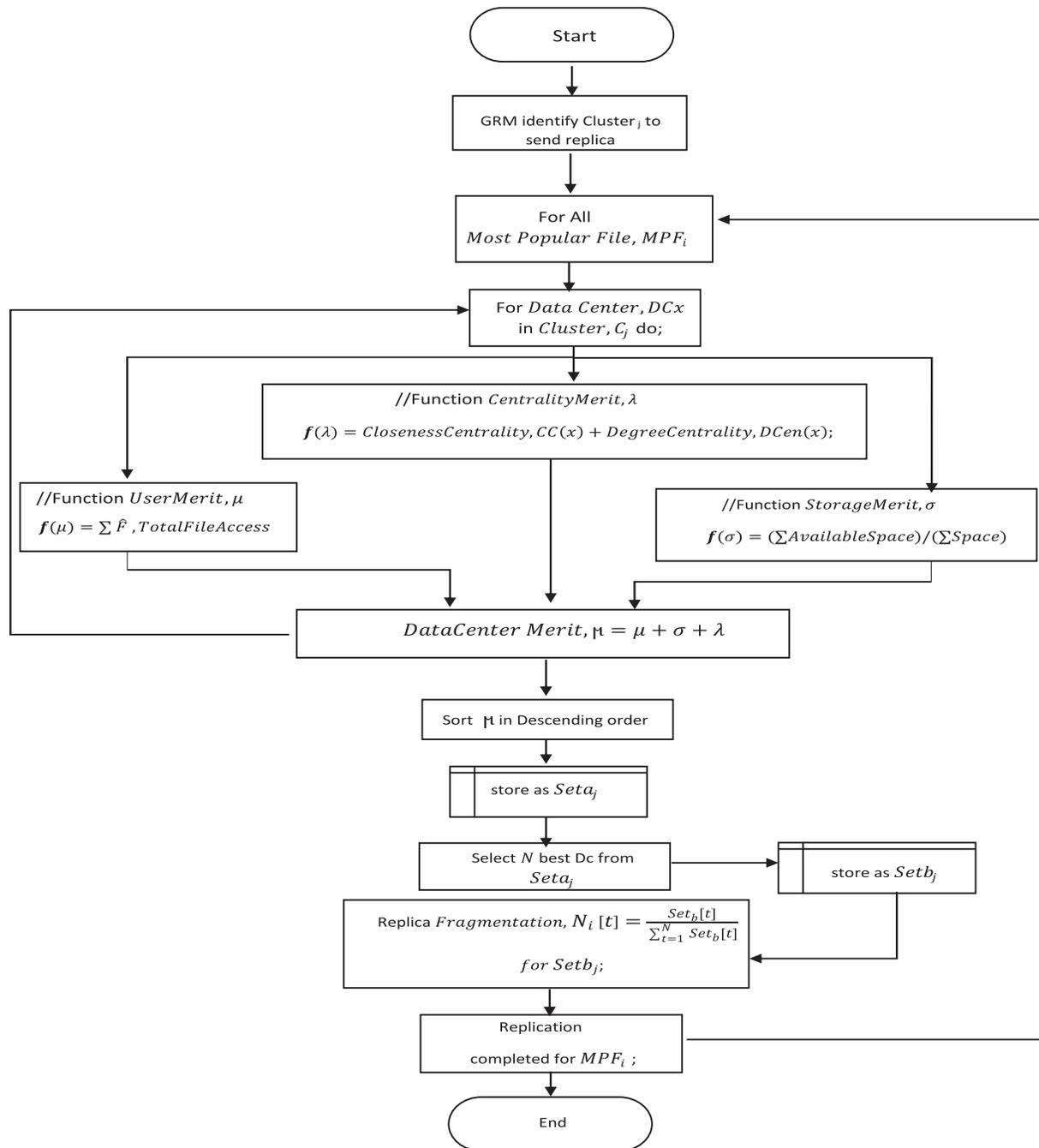


Figure 2: RS-DCSM Flowchart

4.1 Effective Network Usage (ENU)

This study measures Effective Network Usage (ENU) to demonstrate the RS-DCSM algorithm's competence to provide further performance while using fewer network resources. The ENU formula

has been adopted from [39] as Eq. (7).

$$ENU = \frac{N_{rfa} + N_{fa}}{N_{lfa}} \tag{7}$$

N_{rfa} in Eq. (7) indicates the number of access times that site reads a file from a remote site (N is number, r is remote, f is file and a is access) which the obtained value is added to the total number of file replication operation referred as N_{fa} (N is number, f is file and a is access) and divided by N_{lfa} denoted as a number of times that site reads a file locally (N is number, l is local, f is file, and a is access). The ENU calculation is normalized to a scale between 0 and 1.

4.2 Replication Frequency (RF)

The number of replications for each data access in a replication environment is measured by *Replication Frequency (RF)*. The lower the value, the more efficient the methods for allocating replicas in storage nodes. The ratio of replication to the frequency of data access is measured by adopting formula from author [39] as Eq. (8) below:

$$RF = \text{Frequency of Replication} : \text{Frequency of Data Access} \tag{8}$$

The *Frequency of Replication* in Eq. (8) denotes the number of replications accomplished in the entire simulation, and *Frequency of Data Access* is referring to the number of data access in the replication system. This parameter is used to determine how many replications are necessary for each data access. As a result, the lower the replication frequency, the better the method introduced, which can reduce heavy network demand and demonstrate appropriate replicas available locally.

Few job iterations were used in the experiments: 100, 300, 500, 700, 900, and 1100 jobs per round. Results for efficient network utilization (ENU) acquired from tests conceived of random file sizes on a scale of 100 Mb to 10,000 Mb are presented in Fig. 3a in this section.

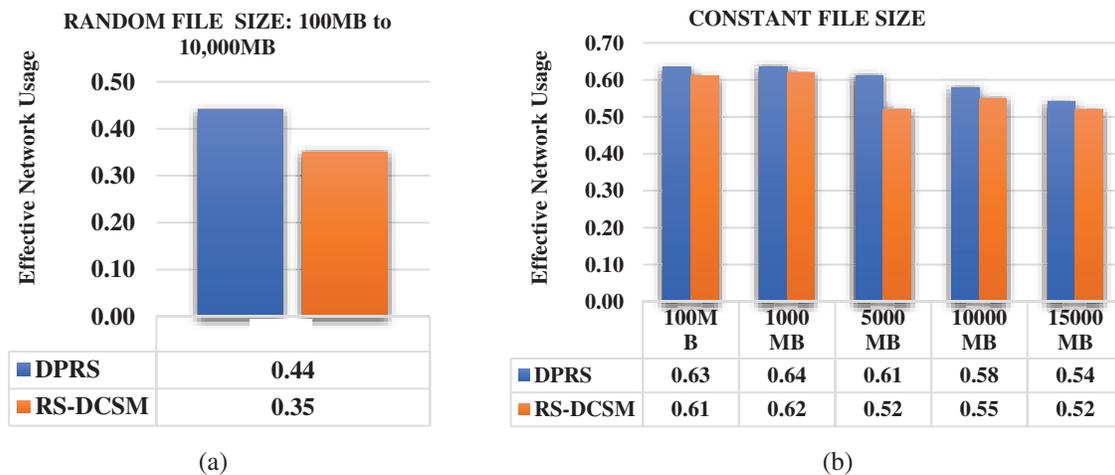


Figure 3: (a) and (b): Network Usage vs. File Sizes

Fig. 3a presents a bar chart for both RS-DCSM and DPRS algorithm results. Eq. (7) is applied to measure network usage in this experiment. As observed in the bar chart, an average of 20% decrement in network usage was obtained by RS-DCSM than the DPRS algorithm which shows better efficient of the network usage. Specifically, DPRS got 0.44 network usage while RS-DCSM used a lower network

with only 0.35. The findings provide strong support for the proposed RS-DCSM, which aims to reduce network load by directing created replicas to the most appropriate sites. As a result, RS-DCSM has the lower ENU result due to its capability to obtain relevant data files locally rather than regularly acquiring replicas from remote sites. DPRS, on the other hand, used more bandwidth since it ignored some of the essential criteria, such as temporal locality. DPRS says that their technique assigns replicas among all sites, but they failed to consider the repercussions of resource waste. The waste is because some data centers have high data access and can request popular files rather than allocating the replica to the data center with the high request for single files, which is not particularly popular among users. As a result, the DPRS algorithm's choice of the data center for replica placement will not be the most popular, resulting in a waste of resources.

In a similar simulation scenario, multiple constant file sizes are induced to observe the method accuracy further. As a result, Fig. 3b shows the ENU findings for various constant file sizes, including 100, 1000, 5000, 10,000, and 15,000 Mb.

Based on Fig. 3b, RS-DCSM was observed to deliver efficient network usage with 4%, 3%, 15%, 5%, and 4% enhancements for 100, 1000, 5000, 10,000, and 15,000 Mb file sizes, respectively, outperformed DPRS by 6% in total average improvement. It means that the RS-DCSM algorithm used less network bandwidth than the DPRS algorithm during the experiment. Despite the method's extensive multi-factors, RS-DCSM also focuses on a common but essential aspect in allocating the mass size of replicas: providing adequate storage in data center selection criteria. The results obtained even with bigger file sizes do not influence RS-DCSM's ability to reduce network utilization. Furthermore, due to the degree of centrality element, RS-DCSM produced better outcomes than DPRS. Allocating segmented data in multiple data centers has a significant risk of access delay when the network is overloaded, as this research suggests segmentation of data files. Therefore, despite the fact that RS-DCSM addresses fault tolerance through introducing the degree of centrality, at the same time, it allows for different ways to speed up replica retrieval. As a result of evaluating the degree of centrality in data center selection, system users benefit from having various paths to get data without waiting in a queue. According to [26,56], and [13], the degree of centrality that addresses fault tolerance improves performance by allowing faster data access and downloads, even if replicas are not accessible locally due to network path failure for unknown causes.

An additional experiment was undertaken to ensure that RS-DCSM has no limitations in other aspects of replication performance to verify its competency further. Hence, replication frequency is evaluated to support this assertion. Subsequently, replication frequency for random and constant file sizes are measured in the same experiment context. The algorithm has established the capacity to allocate replica copies at the best local storages the lower the replication frequency evidence. Thus, analytical graphs are presented as in Fig. 4a for random file sizes and Fig. 4b for constant file sizes.

As in Fig. 4a, DPRS obtained 0.11 replication frequency per data access. Hence, approximately 10 replicas were created for DPRS when 100 data are accessed in a replication environment. Instead, RS-DCSM shows 0.09 replication frequency, which resulting about 9 replicas are generated for 100 data access. The overview shows, both algorithms have close results, yet, RS-DCSM achieved 14% reduced replication frequency on average. The percentage appears to prove it has better capability to reduce the requirement for additional replica placement than DPRS. In conclusion, compared to DPRS, the RS-DCSM algorithm evidenced an adequate number of copies created and accessible at local data centers; conversely, the DPRS imposing a higher number of replica creation to meet local requirements.

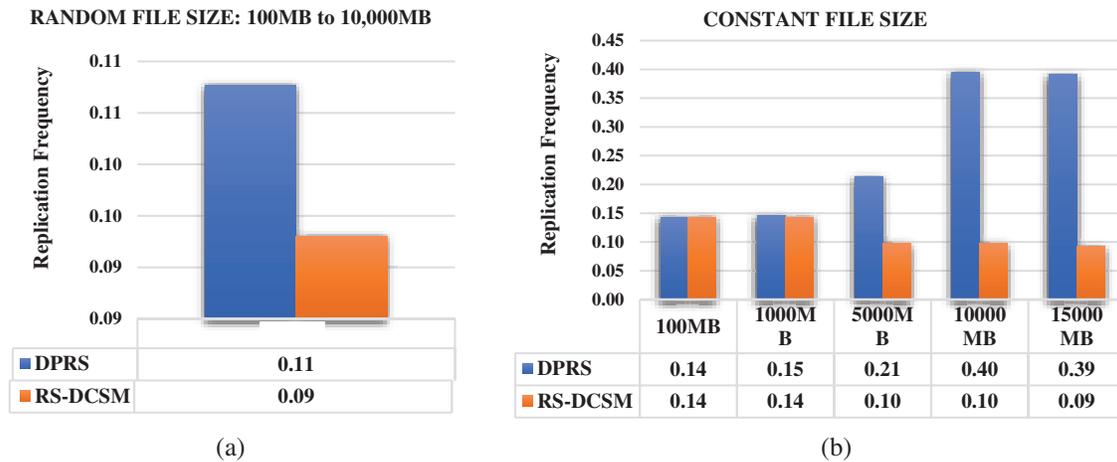


Figure 4: (a) and (b): Replication Frequency vs. File Sizes

Additional experiments were conducted with constant file size measurements, and the findings were compared accordingly. Fig. 4b proves that the RS-DCSM algorithm has a minimal replication process compared to DPRS. Results show that RS-DCSM has outreached the DPRS algorithm to a certain extent of file sizes. At file sizes of 100 and 1000 Mb, the replication frequency for both DPRS and RS-DCSM tolerates similar outcomes, which are fewer than 0.15 replication frequency required per data access, i.e., at least 10 replicas are made per 100 data access. Meanwhile, when analyzing the peak of 5000 Mb file size, RS-DCSM was shown to have created fewer replicas than DPRS, with RS-DCSM preserving nearly the same volume of replica creation with 0.1 replicates per data access. Conversely, DPRS, on the other hand, requires the creation of approximately 20 replicas for every 100 data points examined. RS-DCSM lowered the replica frequency by 55% when larger than 5000 MB files were sent in the simulation scenario.

RS-DCSM, on the other hand, appears to maintain a comparable low replication frequency. At large file sizes of 10,000 and 15,000 Mb, RS-DCSM lowered huge new replica creation requirements by 75% and 76%, respectively. The significant contribution has influenced the improvement exclusively due to the factors absorbed in the RS-DCSM algorithm in determining the best data center to allocate replicas. As an outcome, users require fewer replica copies because the data is always available locally, reducing the requirement to retrieve files remotely and eliminate extra duplicate creations. The high number of replication frequencies for DPRS, on the other hand, derives from the need for additional replication to accommodate for the lack of data available in local data centers. These additional replications are required in DPRS due to drawbacks in the DPRS algorithm’s data center selection method. As a consequence, replica copies are not efficiently distributed in local data centers. A large proportion of remote replica access eventually causes the DPRS algorithm to increase the number of new replicas created, contributing to the high replication frequency.

In conclusion, the graphs illustrate that the RS-DCSM algorithm’s capacity to establish effective network usage does not result in any additional disadvantages in the cloud replication environments. On the contrary, efficient network usage is achieved while replication frequency is maintained by using this adaptive RS-DCSM.

5 Conclusion and Future Works

In a nutshell, this research met its goals while also improving cloud replication performance. The proposed RS-DCSM algorithms outperformed the DPRS algorithm [39], proven by presented experiment findings. Furthermore, the simulation results presented and analyzed in detail demonstrated that the cloud provider and users will both profit from the suggested RS-DCSM algorithm and will be able to reach their desired goals equally.

This adaptive algorithm will always choose the appropriate data center based on comprehensive selection criteria to ensure replicas are placed locally, storage is balanced, and achieve efficient network usage without increasing the execution time. The simulation findings show that data movement between data centers is significantly reduced, resulting in a 14% reduction in overall replication frequency for RS-DCSM and a 20% increase in network usage efficiency over the DPRS algorithm.

As for this research work extension, future researchers are suggested to include replacement techniques in the research scope. Specifically, data replacement during storage is insufficient was not considered in this scope; however, it is one of the substantial areas that can contribute to performance improvement in cloud replication environments.

Acknowledgement: The Universiti Putra Malaysia and the Ministry of Education (MOE) supports this research work. Utmost appreciation and thanks for providing sufficient facilities throughout this research.

Funding Statement: This research was supported by Universiti Putra Malaysia and the Ministry of Education (MOE).

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

- [1] Q. Xia, W. Liang and Z. Xu, "QoS-Aware data replications and placements for query evaluation of big data analytics," in *2017 IEEE Int. Conf. on Communications (ICC)*, Paris, France, pp. 1–7, 2017.
- [2] K. Liu, J. Peng, J. Wang, W. Liu, Z. Huang *et al.*, "Scalable and adaptive data replica placement for geo-distributed cloud storages," *IEEE Transaction Parallel Distribution. System*, vol. 31, no. 7, pp. 1575–1587, 2020.
- [3] S. Nannai John and T. T. Mirnalinee, "A novel dynamic data replication strategy to improve access efficiency of cloud storage," *Information Systems and e-Business Management*, vol. 18, no. 3, pp. 405–426, 2020.
- [4] I. A. Elgendy, W. Z. Zhang, H. He, B. B. Gupta and A. A. Abd El-Latif, "Joint computation offloading and task caching for multi-user and multi-task MEC systems: Reinforcement learning-based algorithms," *Wireless Network*, vol. 27, no. 3, pp. 2023–2038, 2021.
- [5] W. Ding, X. Yu, H. Zhu, Z. Yan and R. H. Deng, "Deduplication on encrypted big data in cloud," *IEEE Transaction Big Data*, vol. 2, no. 2, pp. 138–150, 2016.
- [6] S. Slimani, T. Hamrouni and F. Ben Charrada, "Service-oriented replication strategies for improving quality-of-service in cloud computing: A survey," *Cluster Computer*, vol. 24, no. 1, pp. 361–392, 2021.
- [7] C. B. Tan, M. H. A. Hijazi, Y. Lim and A. Gani, "A survey on proof of retrievability for cloud data integrity and availability: Cloud storage state-of-the-art, issues, solutions and future trends," *Journal of Network and Computer Applications*, vol. 110, no. August 2017, pp. 75–86, 2018.

- [8] N. K. Nivetha and D. Vijayakumar, "Modeling fuzzy based replication strategy to improve data availability in cloud datacenter," in *Int. Conf. on Computing Technologies and Intelligent Data Engineering (ICCTIDE) 2016*, Kovilpatti, India, vol. 1, pp. 1–6, 2016.
- [9] Q. Liu, G. Wang and J. Wu, "Consistency as a service: Auditing cloud consistency," *IEEE Transaction Network Service Management*, vol. 11, no. 1, pp. 25–35, 2014.
- [10] C. Huang, W. Chen, L. Yuan, Y. Ding, S. Jian *et al.*, "Toward security as a service: A trusted cloud service architecture with policy customization," *Journal of Parallel and Distributed Computing*, vol. 149, pp. 76–88, 2020.
- [11] A. Nayyar, "Handbook of cloud computing: Basic to advance research on the concepts and design of cloud computing," in *BPB Publications*, 2019.
- [12] H. Cai, B. Xu, L. Jiang and A. V. Vasilakos, "IoT-Based big data storage systems in cloud computing: Perspectives and challenges," *IEEE Internet of Things Journal*, vol. 4, no. 1, pp. 75–87, 2017.
- [13] A. Shakarami, M. G. Ali, S. Mohammad and M. Hamid, "Data replication schemes in cloud computing: A survey," in *Cluster Computer*, US: Springer, vol. 7, 2021.
- [14] P. Singh, P. Gupta, K. Jyoti and A. Nayyar, "Research on auto-scaling of web applications in cloud: Survey, trends and future directions," *Scalable Computer*, vol. 20, no. 2, pp. 399–432, 2019.
- [15] M. R. Djebbara and H. Belbachir, "Cost function based on analytic hierarchy process for data replication strategy in cloud environment," *Journal of Theoretical and Applied Information Technology*, vol. 96, pp. 2638–2648, 2018.
- [16] F. Xie, J. Yan and J. Shen, "Towards cost reduction in cloud-based workflow management through data replication," in *Fifth Int. Conf. on Advanced Cloud and Big Data (CBD)*, Shanghai, China, pp. 94–99, 2017.
- [17] Y. Shao, C. Li, Z. Fu, L. Jia and Y. Luo, "Cost-effective replication management and scheduling in edge computing," *Journal of Network and Computer Applications*, vol. 129, no. May 2018, pp. 46–61, 2019.
- [18] R. Maheshwari, N. Kumar, M. Shadi and S. Tiwari, "Consensus-based data replication protocol for distributed," *The Journal of Supercomputing*, 2021.
- [19] B. Alami Milani and N. Jafari Navimipour, "A systematic literature review of the data replication techniques in the cloud environments," *Big Data Research*, vol. 10, no. C, pp. 1–7, 2017.
- [20] J. M. Z. Chen, Y. Tian, J. Xiong and C. Peng, "Towards reducing delegation overhead in replication-based verification: An incentive-compatible rational delegation computing scheme," *Information Sciences*, vol. 56, pp. 286–316, 2021.
- [21] H. E. Ciritoglu, T. Saber, T. S. Buda, J. Murphy and C. Thorpe, "Towards a better replica management for hadoop distributed file system," in *IEEE Int. Congress on Big Data (Big Data Congress)*, 2018, pp. 104–111.
- [22] C. Jiang, T. Fan, H. Gao, W. Shi, L. Liu *et al.*, "Energy aware edge computing: A survey," *Computer Communications*, vol. 151, no. 2018, pp. 556–580, 2020.
- [23] S. U. R. Malik, S. U. Khan, S. J. Ewen, N. Tziritas, J. Kolodziej *et al.*, "Performance analysis of data intensive cloud systems based on data management and replication: A survey," *Distributed Parallel Databases*, vol. 34, no. 2, pp. 179–215, 2016.
- [24] N. Mansouri, M. M. Javidi and B. M. H. Zade, "Hierarchical data replication strategy to improve performance in cloud computing," *Frontiers of Computer Science*, vol. 15, no. 2, 2021.
- [25] F. Castro-Medina, L. Rodriguez-Mazahua, M. A. Abud-Figueroa, C. Romero-Torres, L. A. Reyes-Hernandez *et al.*, "Application of data fragmentation and replication methods in the cloud: A review," in *CONIELECOMP 2019-Int. Conf. on Electronics, Communications and Computers*, Puebla, Mexico, pp. 47–54, 2019.
- [26] N. K. Gill and S. Singh, "A dynamic, cost-aware, optimized data replication strategy for heterogeneous cloud data centers," *Future Generation Computer Systems*, vol. 65, pp. 10–32, 2016.
- [27] A. Awad, R. Salem, H. Abdelkader and M. A. Salam, "A novel intelligent approach for dynamic data replication in cloud environment," *IEEE Access*, vol. 9, 2021.
- [28] H. Abbes, T. Louati and C. Cérin, "Dynamic replication factor model for linux containers-based cloud systems," *Journal of Supercomputing*, vol. 76, no. 9, pp. 7219–7241, 2020.

- [29] 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, no. 2, pp. 234–244, 2014.
- [30] A. Cidon, R. Stutsman, S. Rumble, S. Katti, J. Ousterhout *et al.*, "MinCopysets: Derandomizing replication in cloud storage," *Networked Systems Design and Impementation (NSDI)*, vol. 1, pp. 1–14, 2013.
- [31] Z. Zeng and B. Veeravalli, "Optimal metadata replications and request balancing strategy on cloud data centers," *Journal of Parallel and Distributed Computing*, vol. 74, no. 10, pp. 2934–2940, 2014.
- [32] W. Li, Y. Yang and D. Yuan, "Ensuring cloud data reliability with minimum replication by proactive replica checking," *IEEE Transaction Computer*, vol. 65, no. 5, pp. 1494–1506, 2016.
- [33] J. Liu, H. Shen and H. S. Narman, "Popularity-aware multi-failure resilient and cost-effective replication for high data durability in cloud storage," *IEEE Transactions on Parallel and Distributed Systems*, vol. 30, no. 10, pp. 2355–2369, 2019.
- [34] M. K. Hussein and M. H. Mousa, "A Light-weight data replication for cloud data centers environment," *International Journal of Engineering and Innovative Technology (IJEIT)*, vol. 1, no. 6, pp. 169–175, 2012.
- [35] A. Bowers, C. Liao, D. Steiert, D. Lin, A. Squicciarini *et al.*, "Detecting suspicious file migration or replication in the cloud," *IEEE Transaction on Dependable and Secure*, vol. 18, no. 1, pp. 296–309, 2021.
- [36] A. Kaur, P. Gupta, M. Singh and A. Nayyar, "Data placement in era of cloud computing: A survey, taxonomy and open research issues," *Scalable Computer*, vol. 20, no. 2, pp. 377–398, 2019.
- [37] S. Mazumdar, D. Seybold, K. Kritikos and Y. Verginadis, "A survey on data storage and placement methodologies for cloud-big data ecosystem," *Journal of Big Data*, vol. 6, pp. 1, 2019.
- [38] N. Mansouri, "Adaptive data replication strategy in cloud computing for performance improvement," *Frontier Computer Science*, vol. 10, no. 5, pp. 925–935, 2016.
- [39] N. Mansouri, M. K. Rafsanjani and M. M. Javidi, "DPRS: A dynamic popularity aware replication strategy with parallel download scheme in cloud environments," *Simulation Modelling Practice and Theory*, vol. 77, pp. 177–196, 2017.
- [40] H. Khalajzadeh, D. Yuan, B. B. Zhou, J. Grundy and Y. Yang, "Cost effective dynamic data placement for efficient access of social networks," *Journal of Parallel and Distributed Computing*, vol. 141, pp. 82–98, 2020.
- [41] Y. Zhang, X. Nie, J. Jiang, W. Wang, K. Xu *et al.*, "BDS+: An inter-datacenter data replication system with dynamic bandwidth separation," *IEEE/ACM Transactions on Networking*, vol. 29, no. 2, pp. 918–934, 2021.
- [42] S. Karuppusamy and M. Muthaiyan, "An efficient placement algorithm for data replication and to improve system availability in cloud environment," *International Journal of Intelligent Engineering and Systems*, vol. 9, no. 4, pp. 88–97, 2016.
- [43] N. Mansouri, M. M. Javidi and B. Mohammad Hasani Zade, "Using data mining techniques to improve replica management in cloud environment," *Soft Computing*, vol. 24, no. 1, 2019.
- [44] A. Saleh, R. Javidan and M. T. FatehiKhajeh, "A four-phase data replication algorithm for data grid," *Journal of Advanced Computer Science & Technology*, vol. 4, no. 1, pp. 163, 2015.
- [45] D. Sureshpatil, R. V. Mane and V. R. Ghorpade, "Improving the availability and reducing redundancy using deduplication of cloud storage system," in *2017 Int. Conf. on Computing, Communication, Control and Automation (ICCUBEA)*, pp. 1–5, 2018.
- [46] M. Shorfuzzaman, "On the dynamic maintenance of data replicas based on access patterns in a multi-cloud environment," *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 8, no. 3, pp. 207–215, 2017.
- [47] W. Yang and Y. Hu, "A replica management strategy based on MOEA/D," in *Proc. of the 13th IEEE Conf. on Industrial Electronics and Applications (ICIEA)*, Wuhan, China, pp. 2154–2159, 2018.
- [48] S. Y. Sun, W. Bin Yao and X. Y. Li, "DARS: A dynamic adaptive replica strategy under high load cloud-P2P," *Future Generation Computer Systems*, vol. 78, pp. 31–40, 2018.
- [49] U. Tos, R. Mokadem, A. Hameurlain and T. Ayav, "Achieving query performance in the cloud via a cost-effective data replication strategy," *Software Computing*, vol. 25, no. 7, pp. 5437–5454, 2021.

- [50] N. Mansouri, B. Mohammad Hasani Zade and M. M. Javidi, "A Multi-objective optimized replication using fuzzy based self-defense algorithm for cloud computing," *Journal of Network and Computer Applications*, vol. 171, no. October 2019, pp. 102811, 2020.
- [51] 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.
- [52] P. Hage and F. Harary, "Eccentricity and centrality in networks," *Social Networks*, vol. 17, no. 1, pp. 57–63, 1995.
- [53] A. Londhe, V. Bhalerao, S. Ghodey, S. Kate, N. Dandekar *et al.*, "Data division and replication approach for improving security and availability of cloud storage," in *Proc. -2018 4th Int. Conf. on Computing, Communication Control and Automation (ICCCUBEA)*, Pune, India, pp. 1–4, 2019.
- [54] M. E. J. Newman, "Networks: An introduction," 2010, *Oxford University Press. Artif. life.*, University of Michigan and Santa Fe Institute, vol. 80, pp. 1–6, 2009.
- [55] N. Mansouri and M. M. Javidi, "A new prefetching-aware data replication to decrease access latency in cloud environment," *The Journal of Systems & Software*, vol. 144, pp. 197–215, 2018.
- [56] S. J. Nirmala, A. R. Setlur, H. S. Singh and S. Khoriya, "An efficient fault tolerant workflow scheduling approach using replication heuristics and checkpointing in the cloud," *Journal of Parallel and Distributed Computing*, vol. 136, pp. 14–28, 2020.