Ferdowsi University of Mashhad

Journal of Computer and Knowledge Engineering

https://cke.um.ac.ir

انجمن فناوری اطلاعات و ارتباطات ایران

Information and Communication Technology Association of Iran

A Data Replication Algorithm for Improving Server Efficiency in Cloud Computing Using PSO and Fuzzy Systems [*]

Research Article

Mostafa Sabzekar[1], Ehsan Mansouri[2], Arash Deldari[3]

DOI: 10.22067/cke.2023.83351.1090

Abstract: In different scientific disciplines, large-scale data are generated with enormous storage requirements. Therefore, effective data management is a critical issue in distributed systems such as the cloud. As tasks can access a nearby site to access the required file, replicating the desired file to an appropriate location improves access time and reliability. Replicating the popular file to an appropriate site is a good choice, as tasks can get the necessary file from a nearby site. In this research, a novel data replication algorithm is proposed that is consisted of four main phases: 1- determining 20% of commonly used files, 2- computing five conflicting objectives (i.e., average service time, load variance, energy consumption, average response time and cost) 3- finding the near-optimal solution (i.e., suitable locations for new replica) by the PSO technique to acquire a trade-off among the desired objectives. 4- replica replacement considering a fuzzy system with three inputs (i.e., Number of accesses, size of replica and the last access time). The experimental results denote that the proposed replication algorithm outperforms the Profit oriented Data Replication (PDR) and Bee colony-based approach for Data Replication (BCDR) strategies in terms of energy consumption, average response time, load variance, number of connections, Hit ratio, Storage usage, and cost.

Keywords: Cloud computing, Data Replication, Meta-heuristic algorithms, Fuzzy Systems, Power consumption.

1. Introduction

In the contemporary landscape, cloud computing plays a pivotal role in driving a burgeoning array of internet services. Numerous factors have contributed to the transformation of computing systems to cater to diverse industries' requirements, encompassing scientific breakthroughs, storage technologies, escalated utilization of multiple processes, and burgeoning user demands. From an infrastructural perspective, the cloud denotes a distributed and parallel system comprising a cluster of interconnected virtual machines. Its principal objectives revolve around facilitating users to lease, rather than purchase, computing resources that are accessible from any internet-connected location. Generally, service providers offer cloud computing infrastructure, thereby curbing user expenses through service rental [1, 2].

On one hand, the volume and size of content generated in distributed systems continue to surge, while on the other hand, the quantum of data necessitating processing escalates by the second. The acquisition of this data poses a formidable challenge for system and network designers [3–6]. Data replication emerges as a valuable solution to diminish task execution duration by making data accessible to all relevant nodes. Strategic distribution of replicas across the cloud environment not only balances the load but also augments system efficiency. Data replication involves duplicating files across distinct segments of the system, averting disruptions to the workflow in case a file is damaged or inaccessible by relying on an available copy [7, 8].

Accessing cloud computing services mandates the availability of resources. Nonetheless, scarcity of resources within the infrastructure poses challenges, necessitating allocation of distinct computational resources for different processing tasks. Environments leveraging cloud computing can yield substantial advantages for projects necessitating extensive data processing, such as those in astronomy or meteorology domains. However, the colossal data volume poses a formidable challenge, particularly the replication of data. This predicament accentuates the significance of data replication across multiple servers and locations to uphold data integrity and accessibility. Generally, replication methods are employed to bolster system efficiency, with a focus on the 'how' and 'when' of replication, as well as its elimination. A gamut of methods and strategies has been developed to address these questions, all geared toward curtailing execution time. Consequently, each method necessitates the utilization of an iterative algorithm [9–11].

The main contributions of the paper can be listed as follows:

1. Centrality-based Replica Placement: The method introduces a novel approach for selecting the best site for storing replicas based on the centrality factor and the number of accesses. By considering these factors, the proposed strategy aims to reduce access time and improve data retrieval efficiency.

[*] Manuscript received: 2023 July10, Revised, 2023 September 1, Accepted, 2023 September 18.

[1] Corresponding Author: Assistant Professor, Department of Computer Engineering, Birjand University of Technology, Birjand, Iran.
Email: sabzekar@birjandut.ac.ir.

[2] Department of Computer and Technology, Birjand University of Medical Sciences, Birjand, Iran

[3] Assistant Professor, Department of Computer Engineering, University of Torbat Heydarieh, Torbat Heydarieh, Iran

2. Improved Data Center Utilization: The method also focuses on selecting appropriate data centers for specific services. This selection process benefits both customers and service providers by optimizing data center utilization, leading to improved resource allocation and overall system performance.
3. Fuzzy-Based Replica Replacement: To address storage limitations, the paper introduces a replacement strategy that utilizes a fuzzy system to evaluate the value of each replica. Unpopular replicas that are unlikely to be accessed in the future are replaced with more valuable ones, enhancing overall data management efficiency.

2. Background

2.1. Cloud computing service layers

While the architecture of cloud computing may appear straightforward, its effective operation hinges on astute management at the Network layer, facilitating seamless interconnection of systems. As depicted in Figure 1, cloud services are categorized into three overarching types [12–14], as perceived by cloud computing providers:

1. Software as a Service (SaaS) layer: This stratum empowers users to access existing software via a web browser. Notably, users gravitate towards the latest software iterations, and service providers assume the responsibility of delivering updates. Importantly, a customer's utilization of the service remains detached from the hardware's specifications and capabilities, as all computational processes are executed on the server-side.
2. Platform as a Service (PaaS) layer: This tier entails services offered by providers for application development and deployment. A suite of software accessible as services can be integrated into other layers, thereby reducing the necessity for direct placement of numerous programs onto the virtual machine. A notable exemplar includes an operating system that operates within this context.
3. Infrastructure as a Service (IaaS) layer: Within this stratum, resources such as processors, storage, and network components are accessible to end-users through virtualization. Direct control or access to the cloud computing infrastructure is restricted in this service model. A pivotal characteristic of this layer involves the dynamic allocation of resources through virtualization, underscoring its significance.

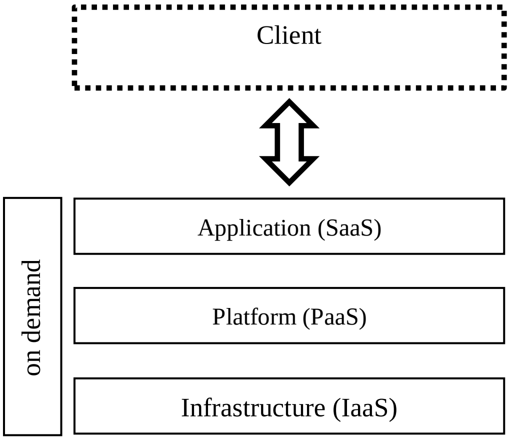


Figure 1. Cloud computing service layers

2.2. Particle Swarm Optimization (PSO) Algorithm

Based on the behaviors of birds and fishes, Particle Swarm Optimization (PSO) represents a population-based method. Groups of particles are characterized by position and velocity vectors, dictating new positions in each iteration. The new position is updated based on particle velocity, particle's best position, and overall best position.

In the PSO algorithm, particle positions are updated using these equations:

$$V_i(t) = w * V_i(t-1) + c_1 * rand_1 * \left(P_{i.best} - X_i(t-1)\right) \\ + c_2 * rand_2 * (P_{g.best} - X_i(t-1)) \tag{1}$$

$$X_i = X_i(t-1) + V_i(t) \tag{2}$$

where, w is the weighted coefficient of inertia, c_1 and c_2 are constant training coefficients, $rand_1$ and $rand_2$ are two random numbers with a uniform distribution in the range of 0 to 1. Furthermore, X_i and V_i are the position vector and the velocity vector of the i-th particle, respectively. The best position found by particle i and thee best position found by the swarm denotes by $P_{i.best}$ and $P_{g.best}$, respectively.

Evolutionary methods have both advantages and disadvantages due to their random nature. Selecting an algorithm is challenging, but evolutionary algorithms have been used successfully for various optimization problems. PSO algorithm offers several advantages [15]:

1. Memory benefit: Past information informs decisions.
2. Particle cooperation: Particles adjust positions based on group conditions, exchanging information to approach the best solution.
3. High convergence: Sharing particle information and quick decisions lead to fast convergence.
4. Implementation simplicity: All stages, from definition to decision-making, are easy to implement without complex math or stats.

2.3. Fuzzy logic System

In uncertain conditions, fuzzy logic is a suitable approach, replacing numerical variables with linguistic analysis. Fuzzy logic deals with values between 0 and 1 and allows verbs like "perhaps to be" or "to be if." Membership in a set is graded, allowing partial membership [16-18]. Notable features of this theory include:

1. Human-like thinking and decision-making simulation
2. Definition of approximations and non-deterministic answers
3. Complex function definition in linear and non-linear forms
4. Simple yet flexible implementation

Figure 2 displays the fuzzy system's architecture and data entry/exit process.

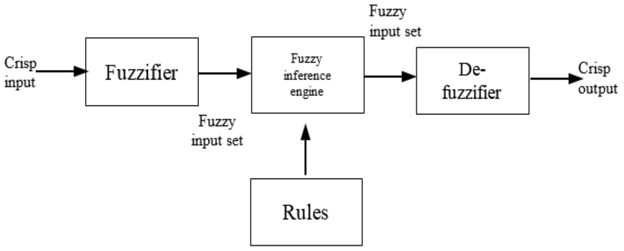


Figure 2. Architecture of a fuzzy system.

As shown in Figure 2, in the first step, all input numbers are converted into fuzzy sets. Then, a fuzzy inference engine evaluates the rules and after collecting the output rules, it is converted into an explicit or numerical value by the de-fuzzifier unit. Finally, the fuzzy results are converted into real numbers.

3. Related works

For data replication in cloud computing, numerous algorithms have been proposed [19], many employing evolutionary techniques [20]. Herein, various algorithms that have enhanced their performance using evolutionary methods at different implementation stages are discussed.

Leo et al. [21] presented a novel strategy using genetic algorithms for new copy replacement in cloud computing. Their approach factors in two main aspects: grouping highly dependent files together to reduce data migration across data centers and considering transfer cost related to file size. The fitness function utilizes total transmission time to determine data transfer amounts. Results show this algorithm reduces data displacement compared to k-means.

Chunlin [22] proposed an algorithm that improves system performance, accounting for file unavailability, data center load, and network transmission costs. It employs a quick sort genetic algorithm to solve the multi-objective copy placement problem. By considering processor capability, memory, disk space, and network bandwidth, this algorithm determines the number and location of copies. The proposed strategy, utilizing a copy transfer approach, outperforms dynamic adjustment strategies (DRAS) [23].

Huang et al. [24] introduced a cost-effective replica placement algorithm under high read/write conditions. It considers storage, update, transmission time, and processing costs. The algorithm determines the number of copies while also finding suitable locations for new versions.

For initializing the population, a heuristic rule based on data support amount and degree is proposed. A hybrid genetic algorithm (HGA) is used, with HGA solutions approaching optimal quality. Navimipour et al. [25] suggest an ant colony strategy to enhance replica selection. Ants choose a center randomly, with subsequent ants attracted to centers with the target file. This strategy significantly reduces access time compared to RTRM by utilizing pheromone information.

Azimi [26] proposed a dynamic data replication algorithm based on the Bee Colony Evolutionary Algorithm (BCDR) for cloud computing environments. The authors considered a hierarchical topology with three levels. There are two levels of connectivity: the first level involves low bandwidth areas and the second level includes LAN (local area network) areas with higher bandwidth connections. The third level includes the sites of each LAN that are connected to each other through high bandwidth. As a result of the proposed algorithm, honey bees stay in a new location if the food area is better or has more nectar than the previous one, and one unit is added to the index. During the search phase, the worker bees determine which sites have the best probability of containing a file. Based on the number of requests, the best site is determined. Based on the evaluation results, the proposed method reduced the execution time compared to LRU, LFU, and BHR [27].

To guarantee the profitability of the cloud service provider, Mokadem et al. [28] proposed a data replication algorithm. The proposed method consists of two main steps. In the first step, the response time is estimated and then compared to a predetermined threshold. In the second step, if the predicted time exceeds the threshold, the supplier is given a new iteration that will provide the maximum profit to the supplier. Based on simulation results, the proposed algorithm reduces file transfer costs by taking into account processor, storage, network, and service costs.

According to the research conducted by Salem et al. [29], an ABC algorithm-based iteration strategy was developed. To determine the optimal copy space, the proposed algorithm first solves the shortest path problem based on the knapsack problem. To achieve a load balance in the system and save the copy from the shortest path at the lowest cost is the main goal of the project. A second step involves implementing an algorithm for finding the optimal sequence of data replication and determining the best path to data centers based on cost. As compared to the strategy (DCR2S) [30] and genetic algorithm (GA), the introduced strategy can reduce data transmission.

Tos et al. [31] introduce a method (PDR) guaranteeing customer performance and cloud service provider profitability. It estimates query response times and profitability-affecting costs to decide if the operation should be repeated.

In [32], The authors introduce a new replication method called hierarchical data replication strategy (HDRS) in this article. The HDRS algorithm involves creating replicas that can increase or decrease based on exponential growth or decay rate, placing replicas based on access load and labeling technique, and replacing replicas based on the future value of the file. The authors compare various dynamic data replication methods using CloudSim simulation and find that HDRS outperforms other algorithms by reducing response time and bandwidth usage. HDRS can efficiently identify popular files and replicate them to the most suitable site, reducing unnecessary replications and balancing site loads to decrease access latency.

The authors in [33] suggested a new replication management strategy called EIMORM, which is an improvement on the MORM algorithm. EIMORM differs from MORM in two ways: it takes into account the cost of replication when placing replicas and assigns weights to data files to determine popular files based on their last access time. The simulation results show that EIMORM can effectively reduce the total cost, particularly for a large number of tasks. However, EIMORM does not address the important step of replica replacement when storage space is full.

The authors in [34], proposed ALO-Tabu algorithm that utilizes a hybrid of ant lion optimization and Tabu search algorithms for solving the replica management problem. The process of selecting the initial population is performed in such way that the algorithm can find better solution.

A comparison of different data replication algorithms' parameters is presented in Table 1. While bandwidth consumption and response time are often prominent, other factors like energy consumption, system load balance, and cost are less emphasized.

Table 1. Parameter comparisons between different data replication algorithms.

Algorithm	Main Idea	Year	Decision making	Replacement	Selection	Cost	Accessibility	Load balancing	Response time	Bandwidth	Energy consumption	Test amount	Metaheuristic method
[21]	Combining similar tasks	2017	-	+	-	-	-	+	+	+	-	5-13 Nodes	GA
[22]	Modeling and using it	2019	-	+	-	+	+	+	+	+	-	1-10 Nodes	GA
[24]	Definition of data classification	2018	-	+	-	+	-	-	+	-	-	10-50 Nodes	EGA
[25]	Accessibility scheme for reading files	2016	-	-	+	-	-	-	+	+	-	5000 Nodes	Ant Colony
[26]	Definition of three-level structure	2019	-	+	+	-	-	-	+	+	-	100-1500 Jobs	ABC
[28]	Profitability of the supplier	2020	+	-	+	+	+	+	+	+	-	500-1500 Nodes	-
[29]	Using the backpack method to solve problems	2019	+	+	+	-	+	+	+	+	-	1-15 Nodes	ABC
[31]	Profitability of the supplier	2016	+	-	+	+	-	-	-	+	-	6 Nodes	-

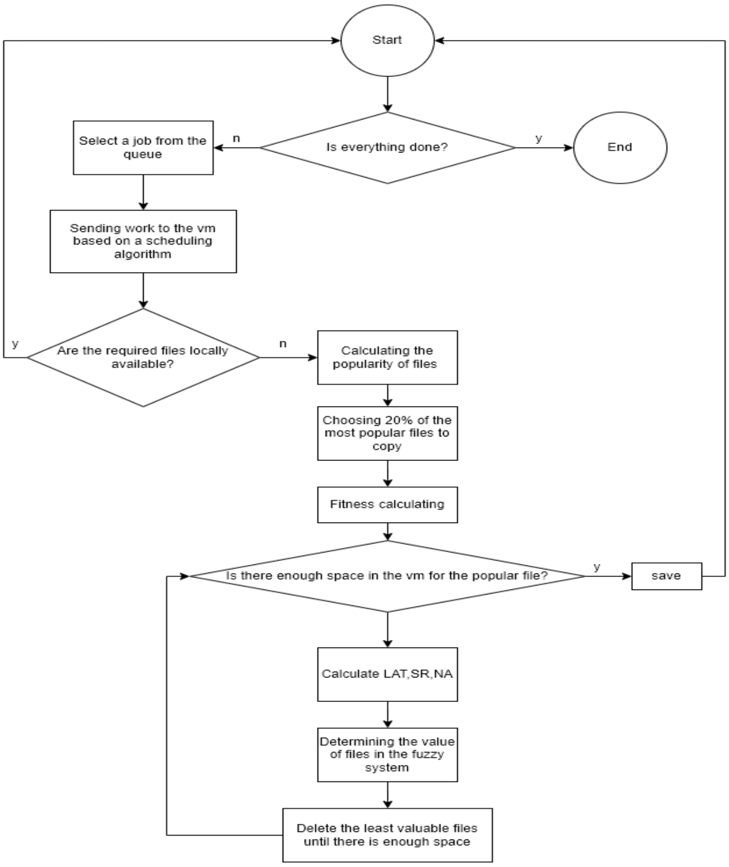

Start
Is everything done?
n
y
End
Select a job from the queue
Sending work to the vm based on a scheduling algorithm
Are the required files locally available?
y
n
Calculating the popularity of files
Choosing 20% of the most popular files to copy
Fitness calculating
Is there enough space in the vm for the popular file?
y
save
Calculate LAT,SR,NA
Determining the value of files in the fuzzy system
Delete the least valuable files until there is enough space

Figure 3. Flowchart of the algorithm

4. The Proposed Method

The algorithm's primary objectives involve determining which files to replicate and when to replicate them. To address this, the algorithm calculates the number of file accesses and the file's popularity. Subsequently, it creates copies for a specific percentage of files based on these factors, with the PSO algorithm guiding the placement of new copies.

To identify candidate files for replication, a value is assigned to each copy, prioritizing less valuable files. This process employs a fuzzy system that takes into account three input parameters: file accesses, copy size, and the time of last access.

The proposed algorithm employs distinct methods for each step, working toward optimal states for individual steps as well as the entire system. Refer to Figure 3 for an overview of the general steps. The algorithm's steps are detailed in the subsequent subsections.

4.1. File Selection and Replication Timing

Cloud system infrastructure constraints, including limited network bandwidth, necessitate careful duplication of files to avoid computational overhead and network congestion. Challenges such as inefficient storage usage, extended task completion times, and security concerns related to user access further underline the importance of an optimal solution. By minimizing the number of files requiring replication, these issues can be mitigated.

To identify less frequently accessed files, the popularity parameter is utilized for selecting the ideal files for replication. This selection process is based on the calculation of each file's popularity, determined using the provided equation:

$$PD_i = (ac_i \times dnc_i) \tag{3}$$

where PD_i represents the popularity of file F_i, ac_i the number of requests, dnc_i the number of data centers that requested the file F_i. Accordingly, the threshold value is determined as follows:

$$T = \frac{\frac{1}{n}\sum_{i=1}^{n} PD_i}{DN_{count}} \tag{4}$$

where the numerator represents the average file popularity and the denominator represents the number of data centers. When a task is assigned to a data center for execution, if the required files are not available locally, the data replication process is applied, In other words, the popularity of the files and the threshold are calculated first, and a certain percentage of the files whose popularity exceeds the replication threshold is selected. A middle limit (20%) is considered in this research based on the availability of capacity and space.

4.2. Locating a New Replica

PSO algorithm finds new replica locations. Key parameters—service time, load variance, energy use, response time, and cost—are fitness functions. PSO helps solve data replication by modeling and determining optimal solutions.

4.3. Locating the New Replication Site

The PSO algorithm is also used to locate new replications exactly. To begin with, four important parameters that are considered fitness functions and have a great impact on cloud computing efficiency. In this research, the average service time, load variance, energy consumption, average response time, and cost are discussed, and then the PSO algorithm is used to determine the best solution to the data replication problem.

4.4. Average Service Time

In order to increase the system throughput of the system, the average service time must be reduced. Reducing the average service time means increasing the processing speed. It is possible to reduce this average time by placing popular files on high performance nodes and less frequently visited files on low performance nodes. Calculating the service time of file *f_i* in data node *j* is as follows [35]:

$$\mathrm{st(i,j)} = \frac{\Phi(\mathrm{i,j}) \times \mathrm{s_i}}{\mathrm{tp_j}} \tag{5}$$

where s_i is the file size of fi and tp_j is the transfer rate of the data node D_j. Each fi file contains r_i copies distributed across different data nodes. We assume that requests from file f_i are modeled as a Poisson function with average access rate A(i).

So, we have:

$$A(i) = \sum_{j=1}^{m} A(i,j) \tag{6}$$

where A(*i,j*) is the access rate of reading requests that are made for the file f_i from the data node D_j. If the file fi is not in D_j node, we set A(*i,j*)=0. The average service time for the f_i file is calculated as follows:

$$\overline{st}(i) = \sum_{j=1}^{m} \left(st(i,j) \times \frac{A(i,j)}{A(i)} \right) \tag{7}$$

The average service time is calculated as follows:

$$MST = \frac{1}{n} \times \sum_{i=1}^{n} \sum_{j=1}^{m} \left(\Phi(i,j) \times s_i / tp_j \times \frac{A(i,j)}{A(i)} \right) \tag{8}$$

4.5. Load Variance

Load variance is used as a measure to show the load balance of the system. In other words, the lower the load variance, the better the load balance in the system. Since the combination of the access rate and the service time of the file f_i gives the exact amount of its load, this load amount *l(i,j)* of the file f_i in the data node D_i will be as follows:

$$l(i,j) = A(i,j) \times st(i,j) \tag{9}$$

The load variance (LV) is calculated as follows [33]:

$$LV = \sqrt{\frac{\sum_{j=1}^{m}\left(l(j)-\bar{l}\right)^2}{m-1}} = \sqrt{\frac{\sum_{j=1}^{m}\left(\sum_{i=1}^{n} A(i,j) \times \frac{s_i}{tp_j} \times \Phi(i,j) - \frac{1}{m} \times \sum_{j=1}^{m}\sum_{i=1}^{n} A(i,j) \times \frac{s_i}{tp_j} \times \Phi(i,j)\right)^2}{m-1}} \tag{10}$$

4.6. Energy Consumption

The total energy consumption is mainly composed of renewable energy consumption (RE) and cooling energy consumption (CE), both of which should be kept as low as possible. In today's world, environmental concerns are one of the biggest challenges, especially in industrialized countries. Among the major factors contributing to pollution of the environment is energy consumption, which should be given a lot of attention and the algorithm should strive to reduce it. Servers' power consumption can be described by a linear relationship between energy consumption and efficiency.

To calculate the renewable energy consumption of the data node D_j which we name in the ERE(j) formula, we will use the following equation [36]:

$$E_{RE}(j) = \sum_{i=1}^{n} \Phi(i,j) \times l(i,j) \times \left(P_{\max}(j) - P_{idle}(j)\right) + P_{idle}(j), \tag{11}$$

where $P_{max}(j)$ shows the maximum power of the data node D_j in the maximum workload and also $P_{idle}(j)$ the power consumption during idle time, so the renewable energy consumption of all nodes is calculated as follows:

$$E_{RE} = \sum_{j=1}^{m} E_{RE}(j) \tag{12}$$

For the same reason, if the outside temperature is 30 degrees and the inside temperature is 20 degrees, we can calculate the total cooling energy consumption as follows [36]:

$$E_{CE} = \sum_{j=1}^{m} E_{CE}(j) \tag{13}$$

where:

$$E_{CE}(j) = E_{RE}(j)/Q, \qquad Q = \frac{1}{\frac{T_{out}}{T_{in}} - 1}$$
$$l(i,j) = A(i,j) \times st(i,j). \tag{14}$$

According to the above, the total energy consumption (EC) can be calculated as follows:

$$EC = \left(1 + \frac{1}{Q}\right) \times \sum_{j=1}^{m}\left(\sum_{i=1}^{n} \Phi(i,j) \times l(i,j) \times (P_{max}(j) - P_{idle}(j)) + P_{idle})\,(j)\right) \tag{15}$$

4.7. Average Response Time

There is an important role to play in reducing latency in any storage system. High bandwidth reduces the amount of delay significantly, so, in this study, we only considered reading delay. Based on the fact that each file has several copies, the average delay ($\overline{L_i}$) is calculated as follows [35]:

$$L_i = \frac{1}{r_i} \times \sum_{j=1}^{m} \Phi(i,j) \times \frac{s_i}{B(j)} \times A(i,j) \tag{16}$$

where $A(i,j)$ is the percentage of read requests sent from the data node D_j to read the file F_i. $B(j)$ is the minimum bandwidth of the data node D_j, so, we will calculate the average delay time for the whole system as follows [36]:

$$ML = \sum_{i=1}^{n} L_i/n = \sum_{i=1}^{n}\left(\frac{1}{r_i} \times \sum_{j=1}^{m} \Phi(i,j) \times \frac{s_i}{B(j)} \times A(i,j)\right)/n \tag{17}$$

4.8. Cost

A common way to replicate data in traditional systems is to create as many copies as possible to maximize the use of resources to increase overall system performance. In cloud systems, the implementation of this method of data replication is not cost-effective for cloud service providers and can lead to excessive and incorrect use of resources and reduced system efficiency. Maintaining an optimal number of copies saves resources and overall costs, especially for servers, so making as many copies as possible is not always the best option.

Several nodes are used by cloud providers to manage and handle user requests. Each of these nodes requires electricity and some hardware to function properly. These items add to the computational costs (C_i). Another cost is related to the use of the network (C_b). The information required by the requests are continuously sent to different parts of the network and to different cloud destinations globally. Consumable memory (C_s) is another cost-related item that the provider pays for each node to provide an empty space for use and placing copies in it. The cost of each of the items varies from one cloud service provider to another, so it is hard to determine which costs more. As a result, the total amount of expenses (ex) is calculated as follows:

$$ex = C_i + C_b + C_s \tag{18}$$

Now, the fitness function is defined as the sum of the above functions (normalized values) and the best place to store a new replica is determined using the PSO algorithm.

Suppose there are n number of files to be copied and there are m data centers, so a position matrix of the number of particles is created which contains the values zero and one. A value of one indicates that the replication should be located in that data center. For example, in Figure 4, file copy 1 (F_1) should be stored in data center 1 (DC_1) and file 2 (F_2) should be stored in data center 3 (DC_3).

	F_1	F_2	F_3	F_4
DC_1	1	0	0	0
DC_2	0	1	0	1
DC_3	0	0	1	0

Figure 4. Example of a position matrix

Now, based on the position matrix, the velocity matrix, is also constructed, and its values are placed in the range as follows:

$$V_k^{i,j} \in [-V_{max}, +V_{max}],$$
$$i \in \{1,2,\dots,n\}, j \in \{1,2,\dots,m\} \quad (19)$$

where V_{max} represents the maximum speed of all particles and V_k represents the speed matrix of the k-th particle. Moreover, Figure 5 shows the velocity matrix for a particle with the position matrix of Figure 4 and the range [-1, 1].

	F_1	F_2	F_3	F_4
DC_1	0.7	-0.3	0.3	-0.8
DC_2	0.2	0.5	0.1	0.43
DC_3	-0.2	0.4	-0.4	0.134

Figure 5. Velocity matrix

It should be noted that p_{best} and g_{best} are two special particles. p_{best} shows the local best particle and g_{best} shows the global best particle among all particles. Note that p_{best} and g_{best} particles are updated in each iteration.

The proposed algorithm evaluates the particles based on the fitness function described before. Therefore, for each particle we have a new fitness function. If a particle's new fitness value is better than that particle's p_{best} fitness value, then p_{best} should be replaced by that particle. In addition, the proposed algorithm uses the p_{best} of all particles and replaces the p_{best} with the current g_{best}, which is better than the current g_{best}.

4.9. Determining the Replica to Be Deleted

If there is not enough free space in the selected data center, the system must delete one or more files to free up space, and after each deletion, update the general copy manager and the local copy manager. The general copy manager includes information about all files in all clusters and the local copy manager includes information about all files in its cluster. In the proposed algorithm, three parameters, number of accesses (NA), replica size (SR) and last copy access time (LAT) are considered.

Considering that the fuzzy system is more accurate and simpler during the decision-making process. Therefore, it is very suitable for decision-making problems with several parameters due to considering the interaction between parameters. For this reason, the use of the fuzzy system for the problem of replacing the copy file has received much attention due to the existing complexities [37, 38].

In the proposed method, a value (RV) is assigned to each copy, and this value is based on the input parameters of the fuzzy system, i.e. copy size, number of accesses, and the last access time of the copy. In the next step, the copies are placed in the list based on their value and in ascending order, so the files at the beginning of the list are candidates to be deleted and free up enough space. Table II contains fuzzy rules in which the different states that exist for determining a file replica. For example, when the number of accesses to the file and the size of the copy and the last access time are high, the overall value of that file is high. On the other hand, if the number of accesses and the file size are low and the last access time is average, the overall value of the file is low and it is placed at the beginning of the list to be deleted.

The fuzzy system used in the proposed method is based on Mamdani system with triangular membership function. There are many membership functions, including Gaussian, etc., but considering that in this problem, many changes are made dynamically in a short period of time, the most suitable function is the triangular function [39].

Considering that linear functions are much easier to design and understand, and also, triangular functions are in the same category. Moreover, due to the dynamic behavior and the need for high speed, the use of these functions will be very useful [40–42].

In general, there are two methods to design a fuzzy system. In the first method, the knowledge of an expert in that field is used, and in the second method, if there is no access to an expert, the learning method can be used. According To the available history of file accesses and the need for only three input parameters, the membership function for the output parameter includes low, medium and high values.

As mentioned earlier, in the fuzzification phase, all input numbers (number of accesses (NA), copy size (SR) and last copy access time (LAT)) after creating input membership functions to the fuzzy set are considered. These steps are shown in Figure 6. De-fuzzification is also shown in Figure 7, where fuzzy rules (if-then rules) are used to process the file, and membership functions are used to convert the fuzzy results to real numbers so that decision can be made.

Table 2. Fuzzy rules

Number of accesses	Copy size	last time the copy accessed	Value
High	High	High	High
High	Middle	High	High
High	High	Middle	High
Low	High	High	Middle
High	Middle	Middle	Middle
Low	Low	Middle	Low
Low	Middle	Low	Low

5. Experimental Results

An 8 GB laptop with an 8th generation Core i5 Intel processor was used to simulate and implement the algorithm. MATLAB 2019 was used to implement the simulation. A description of all the settings and values used to simulate the algorithm can be found in Table III. The number of instructions to execute is between 500 and 4500 Million instructions per second (MIPS) and 100 virtual machines are used to simulate the algorithm.

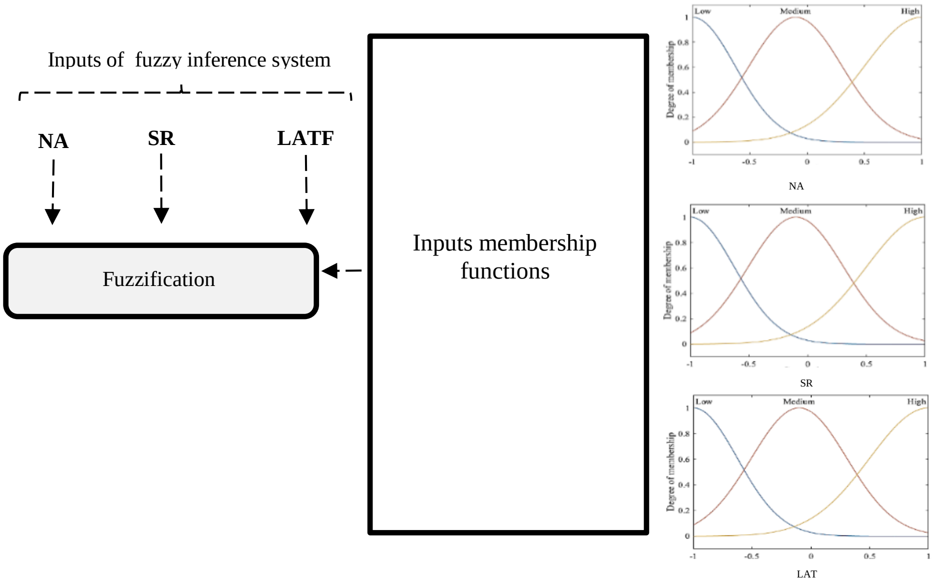
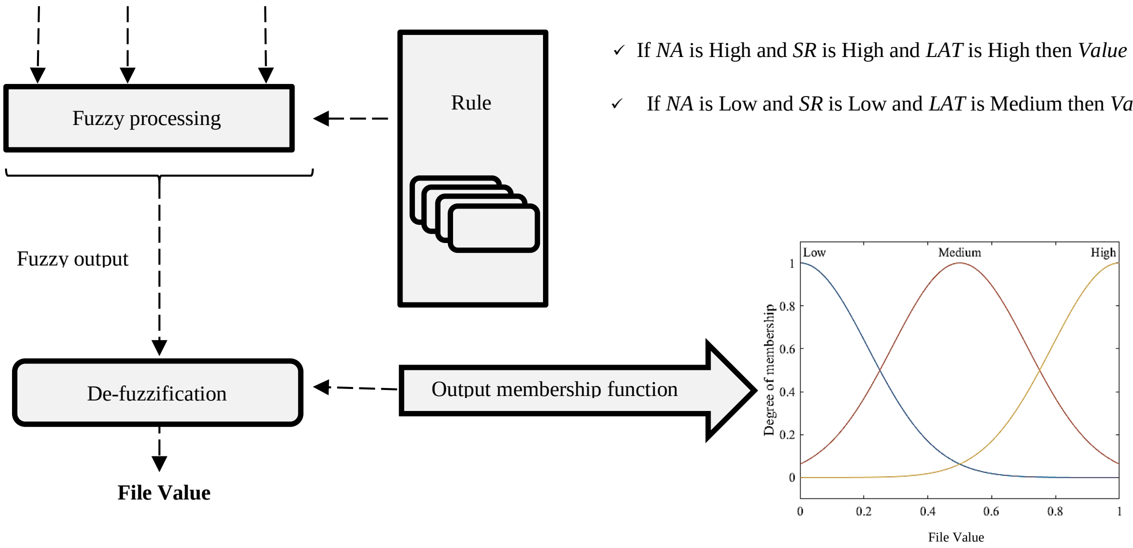


Figure 6. fuzzification steps



- ✓ If *NA* is High and *SR* is High and *LAT* is High then *Value* is High.
- ✓ If *NA* is Low and *SR* is Low and *LAT* is Medium then *Value* is Low.

Figure 7. De-fuzzification steps

Table 4. Parameters used for assessment of the method

Parameter	value
Number of data centers	10-50
Number of virtual machines	100
Number of instructions per second (MIPS)	500-4500
number of processing elements per virtual machine	1-4
Each virtual machine's RAM memory	15-35 GB
Total number of tasks	100-500
Length of each task	100-2000 (MI)
Cost of file transfer	$0.05 per GB
Storage cost	$0.1 per GB
Processing cost	$1 per 106 MI

In the following of this section, we will report and discuss the obtained results with respect to different criteria.

5.1. Average Response Time

A response time is defined as the amount of time it takes from the time the job is sent to receiving the response. The average response time is calculated as follows [43]:

$$AverageResponseTime = \frac{\sum_{j=1}^{m}\sum_{k=1}^{m_j}\left(ts_{jk}(rt) - ts_{jk}(st)\right)}{\sum_{j=1}^{m} m_j} \tag{20}$$

where ts_{jk}(st) and ts_{jk} (rt) indicate the time of sending and receiving task k from/to user j. Moreover, m_j shows the number of jobs of user *j*.

Three different algorithms with different numbers of tasks are shown in Figure 8. Generally, the average response time is considered to be an important efficiency measure for data replication. As can be seen, the average response time naturally increases as the number of tasks increases, but it should be noted that the shorter this time, the more efficient the algorithm is. The experimental results in Fig. 8 indicate that the proposed algorithm has the fastest response time compared to BCDR and PDR algorithms for the number of tasks, reducing the response time by approximately 25% and 17%, respectively, indicating a significant improvement in efficiency and performance. One of the most effective reasons for this improvement is to store the most accessed file in the best data centers.

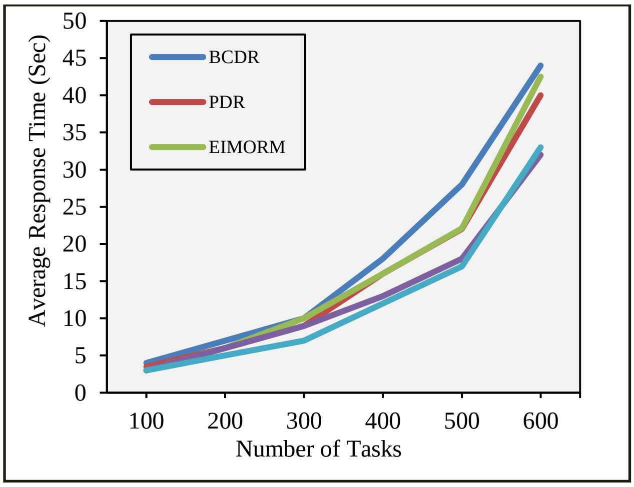


Figure 8 Average response time comparison Energy Consumption

Nowadays, one of the most important issues that has received much attention is the management of energy consumption and trying to reduce the amount of consumption. Considering important issues such as the environment, fuel limitations, high costs, etc. Many experts try to provide various solutions to reduce energy consumption. Most of the existing methods for achieving minimum energy consumption disregard various parameters, which leads to a decrease in system efficiency to some extent.

In the proposed method, as can be seen in Figure 9, the energy consumption for different number of tasks and jobs is always lower than the PDR and BCDR methods. For example, to execute 500 tasks, the proposed algorithm consumes 298 kilojoules of energy, while the results of PDR and BCDR algorithms was 330 and 360 kJ, respectively, and this indicates a 10% and 17% reduction in consumption. One of the reasons for this reduction is the consideration of the energy consumption parameter in the fitness function of the PSO algorithm. Also, other reasons such as increasing workload balance and reducing the number of connections, which will be discussed further, are also effective in reducing energy consumption.

5.2. Load Variance

Usually, load balance in the network is described by a parameter called load variance. Load variance means the standard deviation of data nodes in cloud storage. One of the most important factors influencing the performance of a distributed system is the correct distribution of the load so that the system is in an optimal state. This parameter increases as the number of files increases. In an optimal method, this value should be minimized as much as possible. Figure 10 shows the load variance for different number of files. As can be seen in this diagram, the variance of the load in the HDR algorithm is much lower compared to the PDR and BCDR algorithms, especially for a high number of files. For example, the load variance for the proposed algorithm for the number of 600 files compared to PDR and BCDR has been reduced by 17 and 40%, respectively. The reason for this decrease is that the files are optimally deployed in the data centers, so, the difference in workload on the data centers is also reduced. Considering that in real environments and in the application space, the number of files is generally very large, which clearly shows the proposed algorithm outperforms compared to other methods.

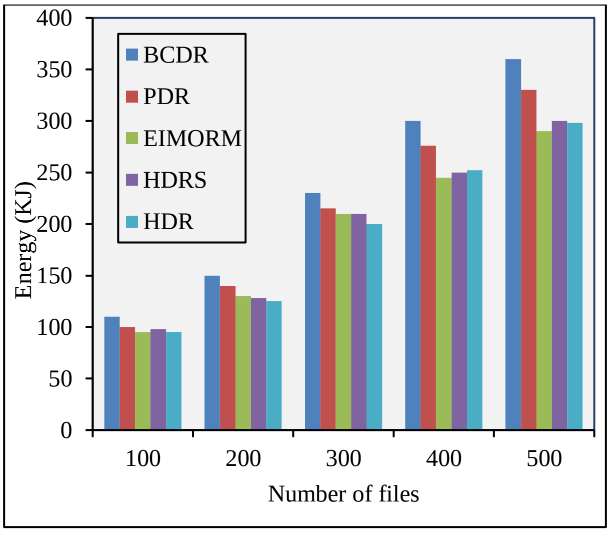


Figure 9. Comparison of energy consumption

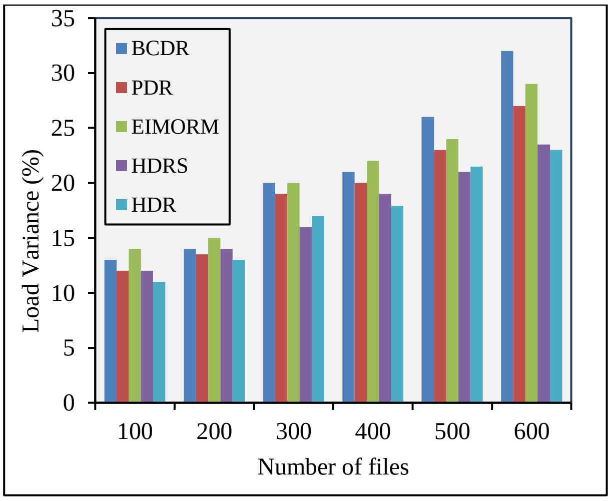


Figure 10. Comparison of load variance

5.3. Number of Connections

Figure 11 shows the number of connections. The number of connections plays an important role in the response time and System efficiency. In other words, reducing the total number of connections, even if it is a small amount, is necessary to reduce data access delay and prevent bandwidth congestion. The HDR algorithm stores the copy file in the best place in terms of time and space, which will reduce the number of connections. The simulations for similar tasks showed 1459, 1530, and 1605 connections for HDR, PDR, and BCDR algorithms, respectively, representing a 5% and 10% reduction in the number of connections.

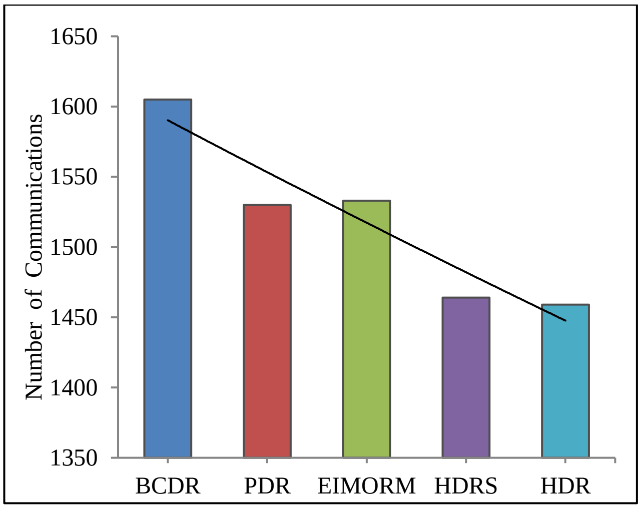


Figure 11. Comparison of load variance

5.4. Hit Ratio

The hit rate is equal to the ratio of the number of local file accesses to the total number of accesses. Total accesses include local file accesses, total number of copies, and total number of remote file accesses. Figure 12 shows the hit rate for 1000 jobs in the compared algorithms. It is easy to understand that the HDR algorithm has the highest rate compared to the PDR and BCDR algorithms. Therefore, in the proposed method, the hit rate has increased by 37% and 57%, respectively, compared to PDR and BCDR algorithms. Because of this increase the number of local accesses to files is increased due to storing copies in the right places and based on the number of accesses to files and creating unnecessary copies, so, the total number of copies and the number of remote accesses to files is reduced ambiguous. The hit rate increases by decreasing the denominator of the fraction.

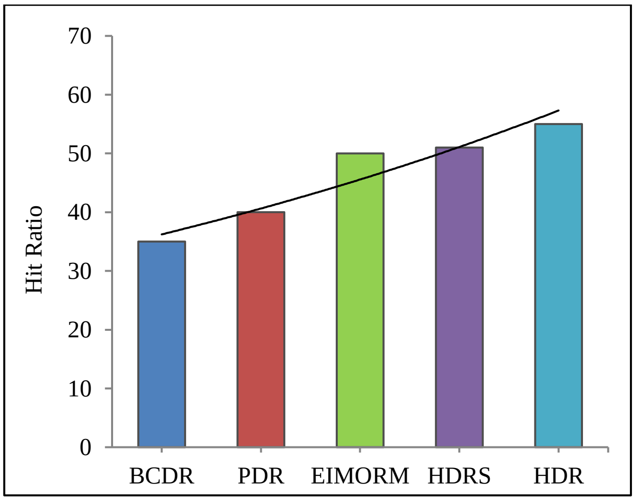


Figure 12. Comparison of hit ratio

5.5. Storage Usage

Storage space is undoubtedly one of the main elements in the cloud, so monitoring the usage of storage resources can provide useful information. This issue can be considered in proposing an efficient replication method from two important perspectives: on the one hand, the goal can be to minimize the storage space consumption, because the cost of resources is proportional to the amount used. On the other hand, the cost may be fixed and the main goal is to maximize the use of storage space.

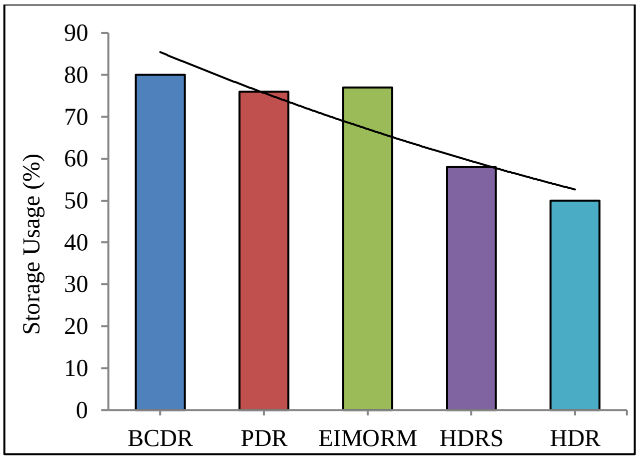


Figure 13. Comparison of storage usage

Figure 13 shows the amount of storage space for the discussed algorithms. As it is clear, in the proposed algorithm, a reduction of 38 and 34 percent of storage space consumption has occurred, respectively, relative to the BCDR and PDR algorithms, which is a significant reduction. The reason for this decrease is two basic things: firstly, considering the file popularity parameter, only popular and frequently used files are copied, so many additional copies cannot be done, and secondly, the file replacement method is used, that is, files that have less value are deleted. Considering the hardware limitations as well as the high costs of preparing, setting up and maintaining storage spaces and considering the high importance of economic justification in the world of information technology, this reduction in storage space consumption is one of the most important features of the proposed method in this research.

5.6. Cost

As mentioned in the previous parts, one of the important parameters in the use of cloud systems is financial issues. Most of today's researches and solutions try to minimize costs for end users, while one of the important issues in this environment is the costs of cloud providers and improving profitability by reducing their costs. These costs include various things such as space and bandwidth consumption, processing and transfer costs of files, as well as maintenance costs. Therefore, providing a method to guarantee the profitability of cloud service providers along with other benefits for users and creating a balance in this environment will be very useful. As seen in Figure 14, for a certain number of tasks, the cost for PDR and BCDR algorithms is $40 and $46, respectively, while it is $32 for the proposed method, and this represents a 20% and 30% reduction in cost.

It is general. One of the important reasons for this reduction is that in the PDR and BCDR algorithms, the files are not copied in the right place, and when requesting a job, the file does not exist locally, and it is necessary to move the file, which incurs various costs, including It involves transfer and processing, but in the proposed method, due to considering the popularity of the file, the copy files have a suitable distribution, and due to the local presence of the files, this problem is avoided to a large extent.

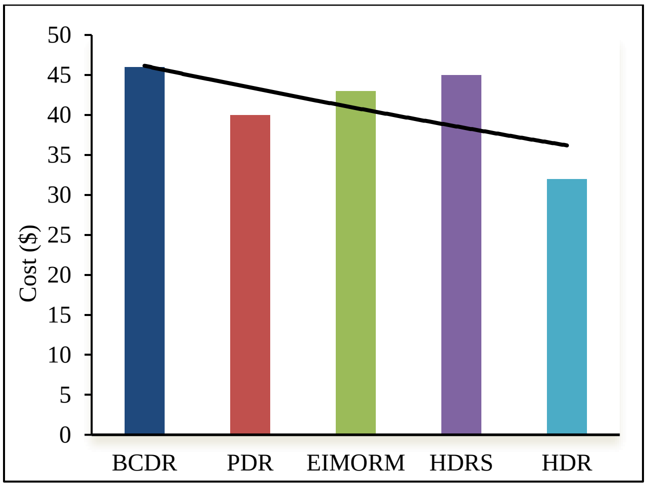


Figure 14. Comparison of Cost

5.7. Average Response Time (ART)

The response time for a datafile is the interval between the submission time of the task and return time of the result. The average response time of a system is the mean value of the response time for all data request tasks of the users. Therefore, for the last experiment, we compare the average response time of the proposed method with others. Figure 15 shows the obtained results.

As shown in Figure 15, the proposed method reports lower ART.

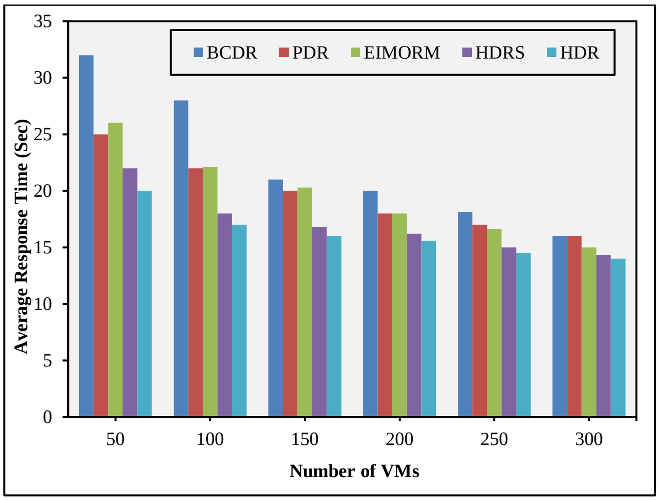


Figure 15. Comparison of average response time

6. Conclusion

The study introduces a novel data replication approach, referred to as HDR, aimed at enhancing data recovery within the cloud environment. The HDR method aims to decrease program response times, lower connection count, elevate hit rates, establish balanced system loads, and minimize storage consumption. By strategically placing copies near desired data centers and minimizing interconnections, the approach achieves load balancing across the entire system, leading to a remarkable 25% reduction in average response time—a pivotal efficiency metric.

In cloud environments, an optimal system should ensure swift response times, a goal accomplished by simultaneously considering parameters such as access frequency, copy size, and last access time in a fuzzy framework. Moreover, system cost-effectiveness is crucial, given the substantial expenses related to hardware and infrastructure. The research places significant emphasis on this aspect, evident in graphical outcomes demonstrating an average storage space reduction of approximately 35%. As storage expenses significantly impact cloud system costs alongside hardware constraints, the adoption of the proposed HDR method emerges as a potent strategy to substantially curtail overall expenditure.

7. References

[1] Chauhan, S., Pilli, E., Joshi, R., and Singh, G., "Govil, M.C. Brokering in Interconnected Cloud Computing Environments: A Survey", *J Parallel Distrib Comput*, Vol .133, pp. 193–209, doi:10.1016/j.jpdc.2018.08.001, 2019.

[2] Qiu, X., Sun, P., and Dai, Y., "Optimal Task Replication Considering Reliability, Performance, and Energy Consumption for Parallel Computing in Cloud Systems", *Reliab Eng Syst Saf*, 215, 107834, doi:10.1016/J.RESS.2021.107834, 2021.

[3] Shojaiemehr, B., Rahmani, A., Qader, N., "Cloud Computing Service Negotiation: A Systematic Review", *Comput Stand Interfaces*,Vol. 55, pp. 196–206, doi:10.1016/j.csi.2017.08.006, 2018.

[4] Huang, K., and Li, D., "MRMS: A MOEA-Based Replication Management Scheme for Cloud Storage System", *International Conference on Communications in China*, 2016.

[5] Moura, D., and Hutchison, D., "Review and Analysis of Networking Challenges in Cloud Computing", *Journal of Network and Computer Applications*, Vol. 60, pp. 113–129, doi:10.1016/j.jnca.2015.11.015, 2016.

[6] Aznoli, F., and Navimipour, N., "Cloud Services Recommendation: Reviewing the Recent Advances and Suggesting the Future Research Directions", *Journal of Network and Computer Applications*, 2017.

[7] Slimani, S., Hamrouni, T., Charrada, F., "Service-Oriented Replication Strategies for Improving Quality-of-Service in Cloud Computing: A Survey", *Cluster Comput*, Vol. 24, pp. 361–392, doi:10.1007/S10586-020-03108-Z/METRICS, 2021.

[8] Bello, S., Oyedele, L., Akinade, O., Bilal, M., Davila Delgado, J., Akanbi, L., Ajayi, A., and Owolabi, H., "Cloud Computing in Construction Industry: Use Cases, Benefits and Challenges", *Autom Constr*, 122, 103441, doi:10.1016/J.AUTCON.2020.103441, 2021.

[9] Mansouri, N., and Javidi, M., "A New Prefetching-Aware Data Replication to Decrease Access Latency in Cloud Environment", *Journal of Systems and Software*,

Vol. 144, pp. 197–215, doi:10.1016/J.JSS.2018.05.027, 2018.

[10] Mansouri, N., "QDR: A QoS-Aware Data Replication Algorithm for Data Grids Considering Security Factors", *Cluster Comput*, Vol. 19, pp. 1071–1087, doi:10.1007/S10586-016-0576-7/METRICS, 2016.

[11] Sun, S., Yao, W., Li, X., "DARS: A Dynamic Adaptive Replica Strategy under High Load Cloud-P2P", *Future Generation Computer Systems*, Vol. 78, pp. 31–40, doi:10.1016/J.FUTURE.2017.07.046, 2018.

[12] Madhubala, R. P., "Survey on Security Concerns in Cloud Computing", Proceedings of the 2015 International Conference on Green Computing and Internet of Things, ICGCIoT 2015 2016, pp. 1458–1462, doi:10.1109/ICGCIOT.2015.7380697.

[13] Tao, M., Ota, K., Dong, M., DSARP: Dependable Scheduling with Active Replica Placement for Workflow Applications in Cloud Computing. IEEE Transactions on Cloud Computing, Vol. 8, pp. 1069–1078, doi:10.1109/TCC.2016.2628374, 2020.

[14] Xie, F., Yan, J., Shen, J., "Towards Cost Reduction in Cloud-Based Workflow Management through Data Replication", Proceedings - 5th International Conference on Advanced Cloud and Big Data, CBD 2017, pp. 94–99, doi:10.1109/CBD.2017.24, 2017.

[15] Marini, F., Walczak, B., "Particle Swarm Optimization (PSO)", *A Tutorial. Chemometrics and Intelligent Laboratory Systems*, Vol. 149, pp. 153–165, doi:10.1016/J.CHEMOLAB.2015.08.020, 2015.

[16] Ojha, V., Abraham, A., Snášel, V., "Heuristic Design of Fuzzy Inference Systems: A Review of Three Decades of Research", *Eng Appl Artif Intell*, Vol. 85, pp. 845–864, doi:10.1016/J.ENGAPPAI.2019.08.010, 2019.

[17] Aubry, P., Marrez, J., Valibouze, A., "Computing Real Solutions of Fuzzy Polynomial Systems", *Fuzzy Sets Syst*, Vol. 399, pp. 55–76, doi:10.1016/J.FSS.2020.01.004 , 2020.

[18] Guillaume, S., "Designing Fuzzy Inference Systems from Data: An Interpretability-Oriented Review", *IEEE Transactions on Fuzzy Systems*, Vol. 9, pp. 426–443, doi:10.1109/91.928739, 2001.

[19] Shingne, H., Shriram, R., "Heuristic Deep Learning Scheduling in Cloud for Resource-Intensive Internet of Things Systems", *Computers and Electrical Engineering*, doi:10.1016/j.compeleceng.2023.108652, 2023.

[20] Zhou, G., Tian, W.H., Buyya, R., Wu, K., "Growable Genetic Algorithm with Heuristic-Based Local Search for Multi-Dimensional Resources Scheduling of Cloud Computing", Appl Soft Comput, doi:10.1016/j.asoc.2023.110027, 2023.

[21] Liu, L., Yang, Y., Wang, H., Tan, Z., Li, C., "A Group Based Genetic Algorithm Data Replica Placement Strategy for Scientific Workflow", *Proceedings - 16th IEEE/ACIS International Conference on Computer and Information Science*, ICIS 2017, pp. 459–464, doi:10.1109/ICIS.2017.7960036, 2017.

[22] Li, C., Wang, Y.P., Tang, H., Luo, Y., "Dynamic Multi-Objective Optimized Replica Placement and Migration Strategies for SaaS Applications in Edge Cloud", *Future Generation Computer Systems*, Vol. 100, pp. 921–937, doi:10.1016/J.FUTURE.2019.05.003, 2019.

[23] Nousias, I., Khawam, S., Milward, M., Muir, M., Arslan, T. A Multi-Objective GA Based Physical "Placement Algorithm for Heterogeneous Dynamically Reconfigurable Arrays", Proceedings - 2007 NASA/ESA Conference on Adaptive Hardware and Systems, AHS-2007, pp. 504–510, doi:10.1109/AHS.2007.8, 2007.

[24] Huang, X., Wu, F., "A Cost-Effective Data Replica Placement Strategy Based on Hybrid Genetic Algorithm for Cloud Services", *Lecture Notes in Business Information Processing*, Vol. 327, pp. 43–56, doi:10.1007/978-3-319-99040-8_4/FIGURES/6, 2018.

[25] Navimipour, N. J., Milani, B. A., "Replica Selection in the Cloud Environments Using an Ant Colony Algorithm", *2016 3rd International Conference on Digital Information Processing, Data Mining, and Wireless Communications, DIPDMWC 2016*, pp. 105–110, doi:10.1109/DIPDMWC.2016.7529372, 2016.

[26] khalili azimi, S., "A Bee Colony (Beehive) Based Approach for Data Replication in Cloud Environments", *Lecture Notes in Electrical Engineering*, Vol. 480, pp. 1039–1052, doi:10.1007/978-981-10-8672-4_80/COVER, 2019.

[27] Park, S. M., Kim, J. H., Ko, Y. B., Yoon, W. S., "Dynamic Data Grid Replication Strategy Based on Internet Hierarchy", *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 3033, pp. 838–846, doi:10.1007/978-3-540-24680-0_133/COVER, 2004.

[28] Mokadem, R., Hameurlain, A., "A Data Replication Strategy with Tenant Performance and Provider Economic Profit Guarantees in Cloud Data Centers", *Journal of Systems and Software*, Vol. 159, 110447, doi:10.1016/J.JSS.2019.110447, 2020.

[29] Salem, R., Salam, M.A., Abdelkader, H., Awad Mohamed, A., "An Artificial Bee Colony Algorithm for Data Replication Optimization in Cloud Environments", *IEEE Access*, Vol. 8, pp. 51841–51852, doi:10.1109/ACCESS.2019.2957436, 2020.

[30] Gill, N. K., Singh, S., "A Dynamic, Cost-Aware, Optimized Data Replication Strategy for Heterogeneous Cloud Data Centers", *Future Generation Computer Systems*, Vol. 65, pp. 10–32, doi:10.1016/J.FUTURE.2016.05.016, 2016.

[31] Tos, U., Mokadem, R., Hameurlain, A., Ayav, T., Bora, S., "A Performance and Profit Oriented Data Replication Strategy for Cloud Systems", *In Proceedings of the 2016 Intl IEEE Conferences on Ubiquitous Intelligence & Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing,* Internet of People, and Smart World Congress (UIC/ATC/ScalCom/CBDCom/IoP/SmartWorld), pp. 780–787, 2016.

[32] Mansouri, N., Javidi, M., Zade, B.M.H., "Hierarchical data replication strategy to improve performance in cloud computing", Front. Comput. Sci, Vol. 15, 152501. https://doi.org/10.1007/s11704-019-9099-8, 2021.

[33] Edwin, E. B., Umamaheswari, P., Thanka, M. R., "An efficient and improved multi-objective optimized

replication management with dynamic and cost aware strategies in cloud computing data center", Cluster Comput 22 (Suppl 5), pp. 11119–11128. https://doi.org/10.1007/s10586-017-1313-6, 2019.

[34] Zade, B., Mansouri, N., Javidi, M. M., "A new hyper-heuristic based on ant lion optimizer and Tabu search algorithm for replica management in cloud environment", *Artif Intell Rev.*, Vol. 56, pp. 9837–9947. https://doi.org/10.1007/s10462-022-10309-y, 2023.

[35] Sun, D. W., Chang, G. R., Gao, S., Jin, L. Z., Wang, X. W., "Modeling a Dynamic Data Replication Strategy to Increase System Availability in Cloud Computing Environments", *J Comput Sci Technol*, Vol. 27, pp. 256–272, doi:10.1007/S11390-012-1221-4/METRICS, 2012.

[36] Long, S. Q., Zhao, Y. L., Chen, W., "MORM: A Multi-Objective Optimized Replication Management Strategy for Cloud Storage Cluster", *Journal of Systems Architecture*, Vol. 60, pp. 234–244, doi:10.1016/J.SYSARC.2013.11.012, 2014.

[37] Singh, H., Gupta, M. M., Meitzler, T., Hou, Z. G., Garg, K.K., Solo, A. M. G., Zadeh, L. A., "Real-Life Applications of Fuzzy Logic", *Advances in Fuzzy Systems 2013*, doi:10.1155/2013/581879, 2013.

[38] Hu, J., Chen, P., Chen, X., "Intuitionistic Random Multi-Criteria Decision-Making Approach Based on Prospect Theory with Multiple Reference Intervals", *Scientia Iranica*, Vol. 21, pp. 2347–2359, 2014.

[39] Bai, Y., Wang, D., "Fundamentals of Fuzzy Logic Control — Fuzzy Sets, Fuzzy Rules and Defuzzifications", *Advances in Industrial Control*, pp. 17–36, doi:10.1007/978-1-84628-469-4_2/COVER, 2006.

[40] Pedrycz, W., "Why Triangular Membership Functions?" *Fuzzy Sets Syst*, Vol. 64, pp. 21–30, doi:10.1016/0165-0114(94)90003-5, 1994.

[41] Herva, M., Franco-Uría, A., Carrasco, E. F., Roca, E., "Application of Fuzzy Logic for the Integration of Environmental Criteria in Ecodesign", *Expert Syst Appl*, Vol. 39, pp. 4427–4431, doi:10.1016/J.ESWA.2011.09.148, 2012.

[42] Gulati, S., Pal, A., "Tuning Fuzzy Logic Controller with SGWO for River Water Quality Modelling", *Mater Today Proc*, Vol. 54, pp. 733–737, doi:10.1016/J.MATPR.2021.10.467, 2022.

[43] López-Pires, F., Barán, B., "Many-Objective Virtual Machine Placement", *J Grid Comput*, Vol. 15, pp. 161–176, doi:10.1007/S10723-017-9399-X/METRICS, 2017.