Journal of Algorithms and Computation

journal homepage: http://jac.ut.ac.ir

A Review of Replica Replacement Techniques in Grid Computing and Cloud Computing

Najme Mansouri*[1] and Mohammad Masoud Javidi †[2]

[1,2]Department of Computer science, Shahid Bahonar University of Kerman, Kerman, Iran.

ABSTRACT

A data-intensive computing platform, encountered in some grid and cloud computing applications, includes numerous tasks that process, transfer or analysis large data files. In such environments, there are large and geographically distributed users that need these huge data. Data management is one of the main challenges of distributed computing environment since data plays on devoted role. Dynamic data replication techniques have been widely applied to improve data access and availability. In order to introduce an appropriate data replication algorithm, there are four important problems that must be solved. 1) Which file should be replicated; 2) How many suitable new replicas should be stored; 3) Where the new replicas should be placed; 4) Which replica should be deleted to make room for new copies. In this paper, we focus particularly on replica replacement issue which makes a significant difference in the efficiency of replication algorithm. We survey

ARTICLE INFO

Article history:
Received 18, February 2019
Received in revised form 25, october 2019
Accepted 17 November 2019
Available online 31, December 2019

Keyword: Cloud computing, Grid, Replica replacement, File popularity, Simulation

AMS subject Classification: 68.

*Corresponding author: N. Mansouri. Email: najme.mansouri@gmail.com
†javidi@mail.uk.ac.ir

1 Abstract continued

replica replacement approaches (from 2004 to 2018) that are developed for both grid and cloud environments. The presented review illustrates the replica replacement problem from a technological and it differs significantly from previous reviews in terms of comprehensiveness and integrated discussion. In this paper, we present different parameters involved in replacement process and show the key points of the recent algorithms with a tabular representation of all those factors. We also report open issues and new challenges in the area.

2 Introduction

2.1 Distributed Systems

The way we do computing are changed due to rapid growth of the Internet and the availability of powerful computers. In recent decades, distributed computing plays a main role to solve a large scale problem. The classification of distributed computing is presented in Fig. [1]. The trend of systems is toward the use of Peer-to-Peer, utility, cluster, and jungle computing [10].

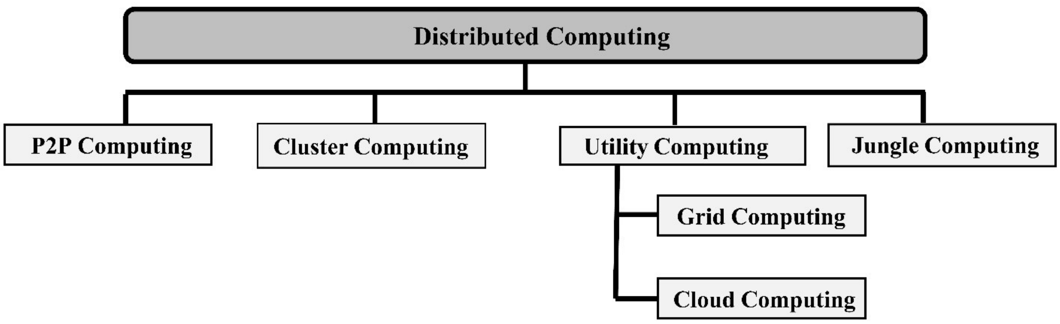

Figure 1: Distributed computing classification.

In a Peer-to-Peer system, every node performs as both a client and a server and presents part of the system resources. Figure [2] indicates that Peer-to-Peer system is self-organizing with distributed management.

Cluster computing is a type of computing that connects several nodes through fast local area networks. Performance and fault tolerance are two main reasons for providing a cluster instead of a single computer [24]. Figure [3] indicates the architecture of cluster computing.

Utility computing provides computing resources and infrastructure for users as needed, and consumers pay service providers according to their usage. The utility computing is generally the grid and the cloud computing that are the recent hot topic of researches.

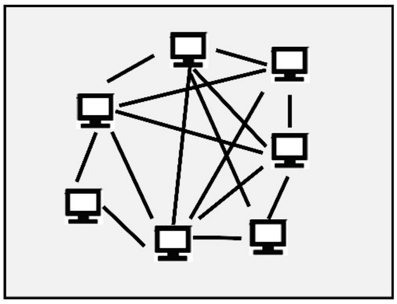

Figure 2: Peer-to-Peer system.

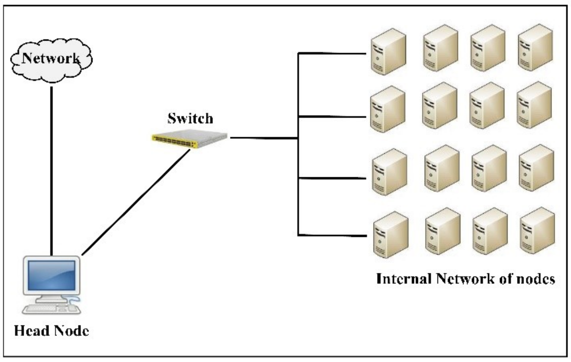

Figure 3: Cluster computing.

Grid computing refers to cooperation of multiple clusters that are loosely coupled and are geographically distributed [3]. Figure 4 shows an overview of grid environment.

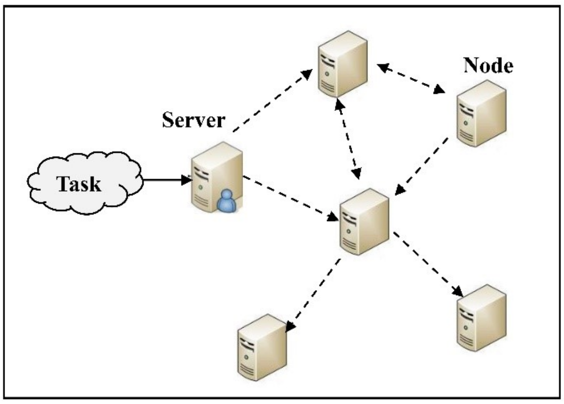

Figure 4: Grid computing.

Cloud computing is not a completely new concept. It is developed based on the grid computing paradigm, and other computing approaches like utility computing and cluster computing [25]. Figure 5 shows an overview of cloud environment. Jungle computing refers to concurrent use of heterogeneous and distributed computing such as clusters, grids, clouds, independent computers, and more [10]. Figure 6 shows the jungle computing. Cluster computing, grid computing and cloud computing are similar in some aspects,

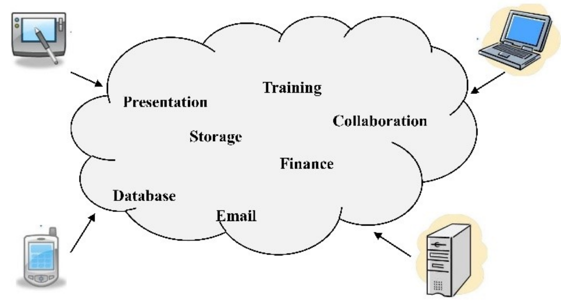

Figure 5: Cloud computing.

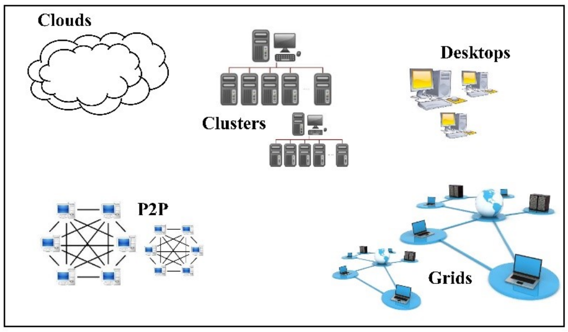

Figure 6: Jungle computing.

while there are some differences among these three technologies that shown in Table 1, [24].

Table 1: Comparison among cluster computing, grid computing and cloud computing.

Cluster Computing	Grid Computing	Cloud Computing
Tightly coupled	Loosely coupled	Dynamic computing
Centralized management	Distributed management	Self-service
The components of system perform similar to a single system	Every component is autonomous	Every component performs as an independent entity
More than two systems are connected to do a project	A large problem is divided among several systems to solve it	Generally, small applications run at the same time.
Single ownership	Multiple ownership	Single/ Multiple ownership
Centralized user management	Decentralized user management	Centralized user management/ third party
Centralized resource management	Distributed resource management	Centralized/distributed resource management
Limited failure management	Limited failure management	Strong failure management

In this work, we focus on cloud and grid computing and so these environment have been described in detail. Figure 7 indicates the architecture of grid computing and cloud computing. Fabric layer contains different resources such as storage, network, and processors. Connectivity layer provides necessary security and communications. Collective layer in grid coordinating various resources. Application layer includes applications of user which execute in virtual organization environments [21].

Cloud providers present three fundamental models as follows [12].

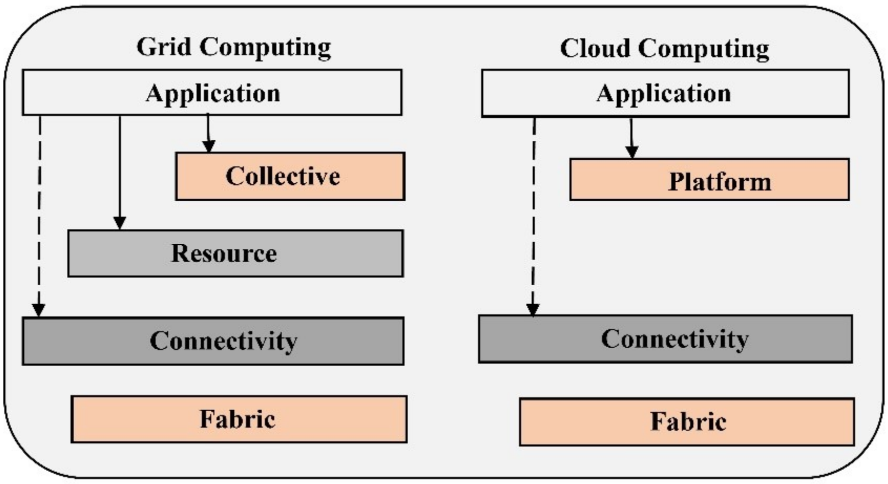

Figure 7: Architectures of cloud and grid computing.

- IaaS (Infrastructure as a Service): Cloud providers provide various infrastructures for processing, storing, and networking such as virtual servers and storages. Therefore, users don't need to handle these facilities except setting of some components such as host firewalls.

- PaaS (Platform as a Service): Cloud providers present a development environment based on the requirement of users. Therefore, users don't need to install necessary software and worry about updating and purchasing the required software such as database.

- SaaS (Software as a Service): Cloud provider presents different types of applications and software for users. Therefore, user can pay some amount to the providers and use these software and applications with interfaces such as web browsers.

Generally cloud infrastructures are classified into four categories according to their scope of usage and methods of deployment [26].

- Private cloud: It refers to the he clouds that infrastructures are operated exclusively for an organization and hence gives high security compared to the other types of clouds. Three main features of private clouds are: (i) high security; (ii) Dedicated resources; and (iii) better customization.

- Public cloud: In public cloud, multiple users can use the cloud resources and services on the common environment. Some benefits of public cloud are: (i) high scalability; (ii) Easy and flexible setup; (iii) Pay-Per-Use.

- Hybrid Cloud: Combination of various private or/and public cloud providers presents a private cloud. Therefore, an organization can run their sensitive applications in private environment and other normal applications are placed in public cloud.

- Community Cloud: It provides exclusive use for the organizations that have common objectives and security requirements. Therefore, establishing cost of community cloud is lower than personal private cloud.

In Fig. 8, we can see five fundamental characteristics of cloud environment as rapid elasticity, measured services, on-demand self-service, resource pooling, and board network access.

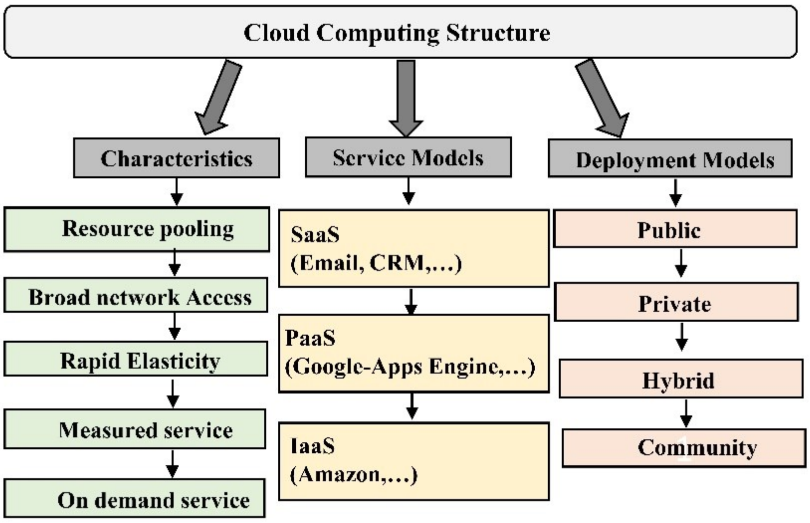

Figure 8: Cloud computing structure.

To the best of our knowledge, this is the first review focusing on replica replacement algorithms for grid and cloud environments. The rest of the review is structured as follows. We start in section 3 by a brief description of the data replication. In section 4, we survey the replica replacement techniques. In addition, we use a tabular representation of all reviewed algorithms based on several main features to facilitate the comparison. Finally, section 5 presents the open research issues of replica replacement.

3 Data Replication

Nowadays, many scientific and applications like High Energy Physics process huge volume of data [15]. In this environment, effective data management is one important challenges. One common solution to effectively solve this issue is to rely on the data replication technique. Data replication can improve response time, throughput, and data availability with storing multiple replicas. Data replication is a hot topic for researchers in the distributed systems such as grid and cloud computing [16,20]. There are four important questions in data replication as follows.

- Which file should be replicated to satisfy users and service provider?

- How many new replicas should be stored to make balance between performance and resource usage?
- Where new replica should be placed to improve network latency and response time?
- Which replicas should be deleted to provide enough space for storing new replica?

Table 2: Data replication surveyed contributions.

Reference	Year	Environment	Main goal
Amjad et al. [2]	2012	Grid	Present a general discussion on data replication strategies for grid based on the nature.
Kingsy Grace et al. [11]	2014	Grid	Compare replica placement and selection techniques.
Tos et al. [32]	2015	Grid	Study the impact of grid architecture on the performance of data replication.
Hamrouni et al. [7]	2016	Grid	Investigate data miningbased replica selection methods.
Malik et al. [33]	2015	Cloud	Discuss data replication techniques that tackle the resource usage and QoS provisioning.
Mansouri et al. [17]	2017	Grid	Provide a comparison of replication algorithms based on the attributes are assumed.

Generally, data replication is classified into two types. Static replication algorithm stores static number of replicas in predefined locations until users delete them. While dynamic replication approach adapts with the change in access pattern of users and status of system during replication process [17,18]. There are several review articles about data replication as listed in Table 2. The novelty of the work we present here, in relation to other surveys, is to focus on a study on replica replacement approaches for both data grid and cloud commuting environments.

4 Replica Replacement Algorithms

The last decade, many replica replacement algorithms have been proposed for grid and cloud environment. Tables 3-6 summarizes the reviewed replacement techniques. Teng et al. [30] combined prediction parameter and replacement cost parameters in the proposed replacement strategy. The proposed strategy predicts the popularity of replica based on the concept of content similarity to store hot spot replicas and hence mean job time is reduced. In addition, it considers cost parameters like bandwidth of network and replica size. Simulation results indicated that the combined replacement algorithm could provide an appropriate balance between job execution time and resource usage. Zhao et al. [36] introduced replica replacement strategy that assigns a weight to replicas according to the number of accesses in the future, the access bandwidth and the file size. The proposed algorithm uses a Zipf-like distribution for predicting file popularity in next time interval. In addition, it considers bandwidth as main factor in replacement process since too much bandwidth usage may block the network. Therefore, the proposed strategy calculates the weight of each replica based on Eq. (1).

$$w(f) = N(f,t,T) \times \frac{S(f)}{B(f)} \tag{1}$$

Where $N(f,t,T)$ indicates the prediction of number of accesses for file f based on Zipf distribution. $S(f)$ is size of file and $B(f)$ shows the mean of bandwidth of all the replicas in system. If the weight of new replica is higher than the replica in the local storage, that replica will be removed and the new replica will be stored into the local site. If enough storage is not available yet, the replacement process will continue. Experiment results demonstrated that the proposed strategy could perform better than Least Recently Used (LRU) and Least Frequency Used (LFU). Liu et al. [13] proposed a replacement

Table 3: Comparison of replica replacement algorithms.

Reference	Teng et al. [30]	Zhao et al. [36]	Liu et al. [13]	Jiang et al. [8]
Year	2005	2008	2011	2010
Environment	Grid	Grid	Grid	Gird
Parameters	Number of requests for file in future, size of file, bandwidth	Number of accesses in future, mean bandwidth, size of file	Number of accesses	Number of accesses
Main idea	Combine prediction and cost factors	Predict file popularity with Zipf law	Consider Inter-reference Recency (IRR)	Use Apriori approach
Simulator used	OptorSim	OptorSim	OptorSim	OptorSim
Evaluation metrics	Storage usage, mean job time	Effective network usage, mean job time	Effective network usage, mean job time	Effective network usage, mean job time, number of remote file access
Compared with	LRU, LFU	LRU, LFU	LRU, LFU, MFU, MRU	LFU

method to improve the performance of data access in grid environment. The proposed strategy stores track of the history information of each replica by Inter-reference Recency (IRR). The number of other replicas accessed between last and penultimate (second-to-last) references of a replica is defined as Inter-reference of a replica. But the number of other replicas accessed from last reference to the current time is referred to the Recency. The authors indicated that if the IRR of a replica is large, the IRR of that replica in the near future is to be large again. Therefore, the proposed strategy selects replica with the largest IRR for deletion since that replica has a low probability of access in next time. The experiment proved that the proposed replacement strategy has much better performance compared with LRU, LFU, Most Frequently Used (MFU) and Most Recently Used (MRU) methods in most cases, particularly for sequential access pattern.

Jiang et al. [8] designed a novel replica replacement strategy based on the Apriori approach for grid environment. The proposed method studies the associated access behaviors of each data intensive job. It considers the accessed files as items in grid environment. It defines the following rules for replacement process.

- When the frequency of set of files is lower than the minimum support then all files in that set should be deleted.
- It replaces itemset that has the least frequency with high priority one.
- If some itemsets have the same frequency then itemset with the smallest size will be selected for replacement.
- The data file that has the lowest value of confidence will be removed in that itemset.

Simulation results indicated that the proposed strategy could reduce number of remote file accesses.

Table 4: Comparison of replica replacement algorithms.

Reference	Teng et al. [27]	SudalaiMuthu et al. [28]	Abawajy [1]	Park et al. [9]
Year	2012	2017	2004	2006
Environment	Grid	Grid	Grid	Grid
Parameters	Data-access frequency, free space	Binomial, LFU, LRU	Size of file, fetching cost	Size of file, latency, bandwidth
Main idea	Determine files	Use Binomial prediction	Calculate cost of fetching	Consider sized-based replacement-k
0	future value	0	0	0
Simulator used	OptorSim	OptorSim	Discrete event simulation	OptorSim
Evaluation metrics	Storage usage, mean job time, effective network usage	Hit ratio, mean job time, effective network usage, computational element usage	Hit ratio, byte hit ratio	Hit ratio, byte hit ratio
Compared with	LRU, LFU	Binomial, LFU, LRU	EBR	LRU, Random

Soosai et al. [27] proposed a new replacement method named the Least Value Replacement (LVR). LVR strategy uses the following function for predicting the number of times a file will be accessed in the next n requests according to the past m requests of the history [4]. The compared experiment results demonstrated that LVR strategy could execute jobs in lower time compared with LRU and LFU methods.
SudalaiMuthu et al. [28] presented a hybrid predictive replacement strategy that considers LRU, LFU, and Binomial method [22] to determine the value of replica. The proposed strategy applies the binary vector for Binomial, LFU, LRU to determine the future request

of the replica. Therefore, it assigns one of number range from 1 to 7 to each replica. For example, if replica r is chosen to be removed based on the only binomial prediction, then the vector is $(1, 0, 0)$ and so the value of replica r is 4. If replica r is chosen to be removed based on only LFU, then the vector is $(0, 1, 0)$ and so the value of replica r is 2. If it is determined by LRU and LFU then $(0, 1, 1)$ gives 3. The low value of a replica means that it has low preference to replace.

Abawajy [1] proposed replica replacement algorithm based on locality, size and cost of files to achieve the best performance possible of grid environment. The proposed strategy keeps size of each file and the cost for fetching each file from remote site. The fetching cost is determined based on Eq. (2).

$$Cost(f, v, u) = T_{process}(v) + Latency + \frac{size(f)}{bandwidth(v, u)} \tag{2}$$

Where $T_{process}$ indicates the processing time, $Latency$ shows the latency for transferring file f, $size(f)$ is size of file, and $bandwidth(v, u)$ shows the bandwidth between node v and node u. The proposed algorithm considers local information to find the file that has size equals to or greater than the new replica to be replaced. In other words, it doesn't remove small files that are least recently used and hence it saves many misses. The results of experiments indicated that the proposed algorithm substantially increases hit ratio compared to the economic-based cache replacement (EBR) strategy [6].

Park et al. [9] introduced a new replacement strategy, called SRB-k (size-based replacement-k). The proposed strategy does not use additional resources such as stacks. It replaces replicas without forecasting future requests, it chooses only a minimum number of files to be replaced in the future. If size of new replica is s then SRB-k strategy finds a file in the local site that its size is r. If there is a file with size s then the file is deleted for providing enough space for new replica. Otherwise, SRB-k strategy applies the replica replacement method considering the k value, which is proportional to the size of the new replica. For instance, if the size of new replica is 1000MB and k is 0.1 (or 10%), then the size of k is 100MB. In other words, 10% of the size of the new replica. If there is no file of that size in the local site and k is 0.1. Then q is obtained by summing the size of new replica and k value. So in this example q equals to 1100MB. p is set of files in local site that their size larger than r and smaller than q. Now, SRB-k strategy deletes a file from p set for replacing based on LRU. The results of simulation demonstrated that the proposed algorithm outperforms than LRU and Random when the storage size was large.

Vijayakumar et al. [34] presented a replica replacement for cloud environment based on fuzzy inference system. The designed fuzzy system has three inputs as number of access for replica, cost of replication, last time of replica access. It is obvious that the replica with high number of access and high cost is not suitable for replacement. So fuzzy system assigns a value to each replica and the replica with the lowest value is candidate for deletion. According to the results of performance evaluation, the proposed algorithm provides better fault tolerance than HDFS [37].

SudalaiMuthu et al. [29] proposed a log based predictive replacement strategy for grid environment. The proposed algorithm determines the value of replica based on the number

Table 5: Comparison of replica replacement algorithms.

Reference	Vijayakumar et al. [29]	SudalaiMuthu et al. [31]	Tian et al. [32]	Bsoul et al. [33]
Year	2015	2015	2007	2011
Environment	Cloud	Grid	Grid	Grid
Parameters	Number of access for replica, cost of replication, last time of replica access	Number of access for replica, the last time access, replica size	Popularity, network bandwidth	Number of access for replica, the last time access, replica size
Main idea	Design fuzzy system	Use Log-based predictive	Consider network usage	Compare importance of importance of group with new replica
Simulator used	Matlab	OptorSim	Real environment	Java
Evaluation metrics	Number of replica	Mean job execution time, hit rate, effective network usage, computational element usage	Mean job execution time, mean bandwidth consumption	Total response time, Total bandwidth consumption
Compared with	HDFS	LFU, LRU, Binomial	LRU, LFU	LRU, LFU

of accesses, the last time access, and replica size. Since the number of access in previous time gives an indication of future requests in the system. In addition, the file size has a main role on network bandwidth and storage usage and hence file size is considered as well. Simulation results demonstrated that the log based predictive replacement strategy performs better about 30% than Binomial based replica replacement algorithm in mean job execution time.

Tian et al. [31] presented a two-step replica replacement method to enhance performance of data grid. In the first step, the proposed strategy calculates the value of replica based on the popularity. In the second step, it tries to minimize cost of replacement by predicting the network bandwidth periodically. The authors defined value of replica similar to the popularity in various virtual organizations (VO). A virtual organization considers some sharing rules among a set of individuals. The proposed strategy determines the popularity of file in grid based on the number of requests in a fixed time interval. Then, it replaces the low value replicas with higher value replicas. Experimental results indicated that the two-step replica replacement method could reduce average bandwidth consumption.

Bsoul et al. [5] introduced a combined replica replacement strategy to enhance data availability in grid. The proposed replication algorithm is an improved version of Fast Spread method. Fast Spread algorithm places a new replica at each site along its path to the requester site. If the storage of site is not enough for new replica, then it deletes a group of files. Fast Spread replication algorithm does not compare importance of new replica and importance of that group of files for replacement process. While the proposed algorithm deletes that group only if the importance of that group is lower than the importance of the new replica. It calculates the value of group based on Eq. (3).

$$GV = \frac{\sum_{i=1}^{n} NOR_i}{\sum_{i=1}^{n} Size_i} + \frac{\sum_{i=1}^{n} NORFSTI_i}{FSTI} + \frac{1}{CuT - \frac{\sum_{i=1}^{n} LT_i}{n}} \tag{3}$$

Where n indicates number of files in the group and $Size_i$ shows the size of file i in the group. NOR_i and $FSTI$ show the number of accesses of file i and the frequency specific time slide, respectively. CuT and LT_i indicate the current time and the last request time of file i, respectively. The simulation results presented that the improved Fast Spread algorithm outperforms than Fast Spread with LRU and Fast Spread with LFU.

Table 6: Comparison of replica replacement algorithms.

Reference Year	Sun et al. [35] 2012	Sashi et al. [23] 2013	Mehraban et al. [19] 2013	Madi et al. [14] 2011
Environment	Cloud	Grid	Grid	Grid
Parameters Main idea	System availability, time of access, failure probability, replica size Consider failure probability	Number of access for replica, replica size, bandwidth Assume a region based network	Popularity, network bandwidth, size of replica Apply Half-life concept	Number of access for replica, time of access Use exponential growth/decay concept
Simulator used	CloudSim	OptorSim	OptorSim	OptorSim
Evaluation metrics	System byte effective rate, response time, successful execution rate	Mean job execution time, storage usage, effective network usage, number of replication	Mean job execution time, storage usage, effective network usage, number of replication	Mean job execution time, hit ratio, effective network usage
Compared with	-	LFU	LRU, LFU	LRU, LFU

Sun et al. [35] proposed a new replication algorithm to improve data availability in cloud environment. The proposed strategy presents a mathematical model to describe the relationship between the number of replicas, system availability, time of access and failure probability of file. It determines the popularity of files based on the history of access and then if the popularity of a file exceeds from a dynamic threshold then replication is triggered. If the storage of selected node is not enough for new replica then it sorts all files in descending order by Replica Factor (RF) and removes the file with the lowest RF. RF is obtained based on Eq. (4).

$$RF_i = \frac{PopF_i}{NR_i \times SF_i} \tag{4}$$

Where $PopF_i$ indicates the popularity of file i that is determined according to the access frequency on time factor. NR_i indicates number of replicas for file i and SF_i shows size of file i. Simulation results proved that the proposed strategy improves task successful execution rate.

Sashi et al. [23] presented a replica replacement algorithm for a region based network. The proposed algorithm predicts the popularity of the file and also takes into account the size of the replica with its bandwidth cost. Popularity of file is defined as access frequency. The cost of replica is obtained as Eq. (5).

$$Cost(f) = AccFrequency(f) \times \frac{Size(f)}{Bandwidth(f)} \tag{5}$$

Where $Bandwidth(f)$ is given by Eq. (6).

$$Bandwidth(f) = \frac{\sum_{i=1}^{n} B_i}{n} \tag{6}$$

Where n is number of replicas for file f. Finally, the proposed strategy deletes replicas that have low popularity and cost. Results of simulation indicated that the proposed strategy improves the network usage and job execution time.

Mehraban et al. [19] introduced a replica replacement algorithm that automatically determines which replica to be removed whenever the storage space of the grid node is full. The presented algorithm enhances the property of the temporal locality by considering frequency of file access, priority file, age, free storage space to find the victim file. It uses the Half-life concept to calculate the popularity of file. Therefore, popularity is obtained as Eq. (7).

$$AF(f) = \sum_{t=1}^{N_T} (acc_f^t \times 2^{-(N_T - t)}) \tag{7}$$

Where NT indicates the number of time intervals that are passed and acc_f^t is the number of accesses for the file f in time interval t. In addition the proposed strategy computes transfer cost based on the size of replica and bandwidth. Now priority replica is determined as Eq. (8).

$$Priority(f) = \alpha \times AF(f) + \beta \times Transfer\_Cost \tag{8}$$

Where α and β are coefficients of the formulas. The proposed strategy deletes replica that has the lowest priority in replacement process. Results of experimental indicated that the proposed replacement algorithm reduces execution time since it deletes victim replicas that have less valuable in the future.

Madi et al. [14] proposed a replica replacement strategy to provide a better grid performance. The proposed strategy assigns a value to each replica based on the exponential growth/decay concept. Value of file can be given as Eq. (9).

$$File\_Value = N_f^t \times (1 + r) \tag{9}$$

Where N_f^t indicates the number of access for file f at time t and r is obtained as Eq. (10).

$$r = (\frac{N_f^{t+1}}{N_f^t}) - 1 \tag{10}$$

Where N_f^{t+1} indicates the number of access for file f at time $t+1$. The proposed algorithm deletes less valuable file to provide sufficient space for storing new replica. Simulation results demonstrated that the proposed strategy could increase hit ratio in grid system.

5 Conclusion

Nowadays, numerous data replication algorithms have been presented to enhance availability of distributed systems. A lot of researches indicate that replication method generally needs to focus on the replica replacement problem. Replica replacement is crucial to the performance of data intensive tasks in cloud and grid. In this paper, we illustrate an in-depth analysis of the main existing replica replacement methods. Some future research issues are summarized as follows. From this review, it can be observed that the number of the proposed replica replacement algorithms is limited compared with that of all works done in replication area and so there is still a lot of ideas to be considered for this problem. Most articles target grid environment, while few target cloud computing. So, in general, few researchers have deeply investigated the benefits of clouds for replica replacement and data intensive scientific applications. Most of the replacement strategies perform according to the single data file prospective, so they pay no attention to the association relationships among files in system. Hence data mining technique can be useful for finding most related files. Another original future direction that we find is to use a meta-heuristic technique to consider several objectives such as failure probability, load balancing, energy consumption and etc. From an experimental point of view, most of replication algorithms are evaluated using simulators and only one work was evaluated on a real distributed environment. Therefore, it is necessary to test methods in a real environment. Moreover, it has been seen that most of replacement algorithms compare their results with some basic strategies such as LRU and LFU. Therefore, extensive experiments are still needed for assessing their evaluation results. In other words, it seems there are shortcomings concerning aspects like evaluation of strategy, size of experiment, and comparison. According to the findings of a literature review from 2004 to 2018, it can be seen that most of the replacement methods do not investigate the impact of various considered factors such as the history length and predefined threshold on obtained results. So, it would be very interesting to study impacts on the obtained results. Furthermore, since the designing replica replacement strategy depends on several parameters so fuzzy inference system can be a good solution. However, it is not suitable to define the fuzzy system with a flat set of rules because the number of rules increases exponentially with the number of parameters. Consequently, rule hierarchy approach is better for this context.

References

[1] Abawajy, J.H., File replacement algorithm for storage resource managers in data grids, Lecture notes in computer science, 3038 (2004) 339-346.

[2] Amjad, T., Sher, M., Daud, A., A survey of dynamic replication strategies for improving data availability in data grids, Future Generation Computer Systems, 28 (2012) 337-349.

[3] Azari, L., Rahmani, A. M., Daniel, H.A. Nasih Qader, N. A data replication algorithm for groups of files in data grids, Journal of Parallel and Distributed Computing, 113 (2018) 115-126.

[4] Bell, W.H., Cameron, D.G., Carvajal-Schiaffino, R., Millar, A. P., Stockinger, K., Zini, F., Evaluation of an economy- based file replication strategy for a data grid, In: 3rd IEEE/ACM International Symposium on Cluster Computing and the Grid, (2003).

[5] Bsoul, M., Al-Khasawneh, A., Eddien Abdallah, E., Kilani, Y., Enhanced fast spread replication strategy for data grid, Journal of Network and Computer Applications, 34 (2011) 575-580.

[6] Carman, M., Zini, F., Serafini, L. K., Stockinger, K., Towards an economy-based optimization of file access and replication on a data grid, In: Proceedings of 2nd CCGRID (2002) 120-126.

[7] Hamrouni, T., Slimani, S., Ben Charrada, F., A survey of dynamic replication and replica selection strategies based on data mining techniques in data grids, Engineering Applications of Artificial Intelligence, 48 (2016) 140-158.

[8] Jiang, J., Ji, H., Xu, G., Wei, X., ARRA: an associated replica replacement algorithm based on Apriori approach for data intensive jobs in data grid, Key Engineering Materials, 439 (2010) 1409-1414.

[9] Jin Park, H., Hoon Lee, C., Sized-based replacement-k replacement policy in data grid environments, In: International Symposium on Parallel and Distributed Processing and Applications, (2006) 353-361.

[10] Kahanwal, B. Pal Singh, T., The distributed computing paradigms: P2P, Grid, Cluster, Cloud, and Jungle, International Journal of Latest Research in Science and Technology, 1 (2) (2012) 183-187.

[11] Kingsy Grace, Manimegalai, R., Dynamic replica placement and selection strategies in data gridsA comprehensive survey, Journal of Parallel and Distributed Computing, 74 (2014) 2099-2108.

[12] Kumari, P., Kaur, P., A survey of fault tolerance in cloud computing, Journal of King Saud University Computer and Information Sciences, (2018).

[13] Liu, W., Shi, F. Du, W., An LIRS-based replica replacement strategy for data-intensive applications, in: 10th International Conference on Trust, Security and Privacy in Computing and Communications, (2011) 1381-1386.

[14] Madi, M., Yusof, Y., Hassan, S. Almomani, O., A novel replica replacement strategy for data grid environment, In: International Conference on Software Engineering and Computer Systems, (2011) 717-727.

[15] Mansouri, N., QDR: a QoS-aware data replication algorithm for Data Grids considering security factors, Cluster Computing, 19 (3) 1071-1087.

[16] Mansouri, N., Javidi, M.M., A new prefetching-aware data replication to decrease access latency in cloud environment, Journal of Systems and Software, 144 (2018) 197-215.

[17] Mansouri, N., Javidi, M.M. A Survey Of Dynamic Replication Strategies For Improving Response Time In Data Grid Environment, AUT Journal of Modeling and Simulation, 49 (2) (2017) 239-264.

[18] Mansouri, N. Adaptive data replication strategy in cloud computing for performance improvement, Frontiers of Computer Science, 10 (5) (2016) 925-935.

[19] Mehraban, M., Khademzadeh, A., Salehnamadi, M.R., A prediction-based replica replacement strategy in data grid, Journal of Basic and Applied Scientific Research, 3(4) (2013) 928-939.

[20] Omer, K., Abdalla, G.M.T., Dynamic Algorithms Replication Using Grid Computing, In: International Conference on Computer, Control, Electrical, and Electronics Engineering, (2018) 1-6.

[21] Samimi, P., Patel, A., Review of pricing models for grid and cloud computing, In: IEEE Symposium on Computers and Informatics, (2011).

[22] Salehnamadi, M. R., Ghaheri, B., New replication method in data grids based on weighted files, Journal of Basic and Applied Scientific Research, 3(7) (2013) 546-553.

[23] Sashi, K., Santhanam, T., Replica replacement algorithm for data grid environment, ARPN Journal of Engineering and Applied Sciences, 8 (2) (2013) 86-90.

[24] Shinde, V., Shaikh, A., Souza, C.D., Study of cluster, grid and cloud computing, International Journal of Advanced Research in Computer and Communication Engineering, 4 (10) (2015) 445-448.

[25] Shuai Zhang, Shufen Zhang, The comparison between cloud computing and grid computing, In: International Conference on Computer Application and System Modeling, (2010) 72-75.

[26] Singh, S., Jeong, Y.S., Park, J.H., A survey on cloud computing security: Issues, threats, and solutions, Journal of Network and Computer Applications,75 (2016) 200-222.

[27] Soosai, A.M., Abdullah, A., Othman, M., Latip, R., Nasir Sulaiman, M., Ibrahim, H., Dynamic replica replacement strategy in data grid, In : 8th International Conference on Computing Technology and Information Management, 2 (2012) 578-584.

[28] SudalaiMuthu, T., RameshKumar, K., Hybrid predictive approach for replica replacement in data grid, In: International Conference on Computer Communication and Informatics, (2017).

[29] SudalaiMuthu, T., RameshKumar, K., a log-based predictive approach for replica replacement in data grid, In: International Conference on Advanced Computing and Communication Systems, (2017).

[30] Teng, M., Junzhou, L., A prediction-based and cost-based replica replacement algorithm research and simulation, In: Proceedings of the 19th International Conference on Advanced Information Networking and Applications, (2005).

[31] Tian, T., Lu, J., A VO-based two-stage replica replacement algorithm, In: International Conference on Network and Parallel Computing, (2007) 41-50.

[32] Tos, U., Mokadem, R., Hameurlain, A., T. Ayav, S. Bora, Dynamic replication strategies in data grid systems: a survey, The Journal of Supercomputing, 71 (11) (2015) 4116-4140.

[33] Tziritas, N., Kolodziej, J., Zomaya, A.Y., Madani, S.A., Min-Allah, N., Wang, L., Xu, C.Z., Marwan Malluhi, Q., Pecero, J.E., Balaji, P., Vishnu, A., Ranjan, R., Zeadally , S., Li, H., Performance analysis of data intensive cloud systems based on data management and replication: a survey, Distributed and Parallel Databases, 34 (2) (2015) 179-215.

[34] Vijayakumar, D., Srinivasagan, K. G., Sabarimuthukumar, R., FIR3: a fuzzy inference based reliable replica replacement strategy for cloud data center, Conference on Computing and Network Communications, (2015) 473- 479.

[35] Wei Sun, D., Ran Chang, G., Gao, S., Zhong Jin, L., Wei Wang, X., Modeling a dynamic data replication strategy to increase system availability in cloud computing environments, Journal Of Computer Science And Technology, 27 (2) (2012) 256 272.

[36] Zhao, W., Xu, X., Xiong, N., Wang, Z., A weight-based dynamic replica replacement strategy in data grids, In: IEEE Asia-Pacific Services Computing Conference, (2008) 1544-1549.

[37] The Apache Software Foundation. Hadoop. http://hadoop.apache.org/core/, 2009.