A DYNAMIC REPLICATION MECHANISM IN DATA GRID BASED ON A WEIGHTED PRIORITY-BASED SCHEME

By

MOHAMMAD SAMADI GHARAJEH

Young Researchers and Elite Club, Tabriz Branch, Islamic Azad University, Tabriz, Iran.

Date Received: 15/04/2019 *Date Revised: 12/05/2019* *Date Accepted: 31/07/2019*

ABSTRACT

Replication is one of the popular tools to determine the availability degree of resources (e.g., data files) in data grids. Since data grids involve limited file storages and high computing costs, replication process is very essential in these networks. This paper proposes a dynamic replication mechanism in a data grid that uses a weighted priority-based replication scheme, called WPRS. It specifies a value for each existing in a local storage based on three parameters including price, number of access time, and present time. When a resource is not available for a desired job, it is hired from other sites in the network. The proposed mechanism removes the file having the least value to increase the free space of data storage. Simulation results show that the proposed replication mechanism surpasses some of the existing replication methods in terms of the number of successful jobs, number of non-successful jobs, and buy price.

Keywords: Data Grid, Data Replication, Weighted Mechanism, Multi-criteria Selection, Priority-based Scheme.

INTRODUCTION

The grid (Hossain, Han, & Poor, 2012; Mahmood et al., 2014; Alkhanak, Lee, Rezaei, & Parizi, 2016; Mishra, Patel, Ghosh, & Mund, 2017; Gharajeh, 2015, 2017; Wang, Jie, & Chen, 2018; Krašovec & Filipčič, 2019) enables efficient computing in scientific applications that is considered as an advanced technology in recent years. It aids researchers to carry out the experiment and simulation processes using potential data resources. There are a large number of applications that use grid infrastructure in resource management (Jinno et al., 2010; Chamola, Sikdar, & Krishnamachari, 2017; Gill, Chana, Singh, & Buyya, 2018). The data grid (Sashi & Thanamani, 2011; Weng, Negi, Faloutsos, & Ilić, 2017) technology is emerged via sharing the data files in grid applications. Various scientific applications can be executed through the speedy and reliable data access by using this technology. Heterogeneous resources of several grid systems can contribute to each other as a unique distributed system in a data grid. If the number of shared data files is more than the file storage capacity and network bandwidth, there will be a major challenge in grid environments (Nicholson, Cameron, Doyle, Millar, & Stockinger, 2008; Wilkinson, 2009; Mariotti, Gervasi, Vella, Cuzzocrea, & Costantini, 2018).

Replication is a well-known distributed strategy in data grids that causes the data availability to be high, the bandwidth traffic to be low, and the fault tolerance to be high. An appropriate replication strategy can enhance the performance of grid systems, increase the number of successful jobs, and reduce the job execution costs (Tang, Lee, Tang, & Yeo, 2006; Mansouri & Dastghaibyfard, 2013; Nazir, Ishaq, Shamshirband, & Chronopoulos, 2018). Since there are a large number of data files in a data grid and there exist a limited bandwidth among grid sites, it is impossible to store all data files on local storages (Rahman, Barker, & Alhajij, 2009). Hence, a replication strategy should be used in data grids to replace old data files by new hired files.

Most of the existing replication methods for data grids use some of the basic parameters such as number of recently used and number of frequently used (Rahman et al., 2009). They cause the number of successful jobs to be decreased, the number of non-successful jobs to be increased, and the buy prices of hired files to be grown considerably. Therefore, it is needed to use some of the dynamic strategies to manage replication process on data grids efficiently. In this paper, a dynamic replication mechanism is proposed for data grids based on a weighted priority-based scheme. When there is not enough space in the local storage to store a requested file exists in other sites, the file having the least value is removed from the local storage. The value of files is calculated based on three important factors including price, number of access time, and present time. It seems that the proposed mechanism can increase the number of successful jobs, decrease the number of non-successful jobs, and reduce the buy prices of hired files.

Grid system is one of the main distributed systems, which accomplishes the most important scientific processes in universities and organizations. Thus, it is needed to present the efficient methods to enhance the performance of these systems. Accordingly, the paper proposes an efficient replication method that can improve the replication process of dynamic grid systems. This method considers price and time as the most important factors to enhance reliability and usability of these systems.

1. Related Works

Least Recently Used (LRU) (O'neil, O'neil, & Weikum, 1993) maintains the information about the pages which have not been used for ages. It claims that the pages which have not been extremely used in the last few instructions will probably be slowly used again in the next times. The LRU replication method removes the page which has the least recently used compared to other pages. The main disadvantages of LRU are that it supports only the fundamental locality principle and cannot be used in a huge database.

The replication method presented in (Ranganathan, Iamnitchi & Foster, 2002) suggests a model for determining the number of replicas in order to keep the availability degree of files in P2P communications. It creates a desired replica for each file exists in data grid. The availability degree of files is measured by their failure rates throughout the network. The main disadvantage of this method is that it does not determine the exact number of replicas.

Least Frequently Used (LFU) (Prischepa, 2004) is a replacement policy that stores the pages on proxy servers based on the frequency number of pages. This number indicates the number of the pages, which have been used by users on the server. If one of the pages should be removed from the server to increase free space of the storage, LFU will remove the page which has the least frequently used by users. The main disadvantage of LFU is that its performance will be reduced when scanning a huge database.

The work presented in (Tang et al., 2006) specifies a popularity rate for each data file based on the data access history. It indicates that the data files which have been heavily used in the past will be used again in the future. The Number of Access (NOA) to each file is calculated by analyzing the data access history. The file which has a NOA value greater than the average NOA will be selected for replication in the network. The main disadvantage of this method is that it replicates only the file having a large NOA without considering the basic replication parameters (e.g., number of recently used).

A replication method in data grids should consider the period of accessed time for each data file. If a file was used in the past even it was not used recently, the file will be probably replicated again. The replication algorithm presented in (Chang, Chang, & Wang, 2008), called Least Access Largest Weight (LALW), considers the accessed time of files on servers. It determines a specific weight for each file having different ages. LALW attaches a tag to the access history record of each file to choose an appropriate file when the replication process is required in the network. The main disadvantage of this algorithm is that it does not consider some of the basic parameters (e.g., number of frequently used). Therefore, it is possible that the file, which has a long accessed time but the least frequently used, will not be removed from the server.

The percentage of accessible files is determined by the accuracy rate of replication location service. If, thereby, the location service is not effective, the number of replica files will be high to ensure data availability. A replication method is presented in (Al Mistarihi & Yong, 2008) to perform replication process based on the accuracy rate of location service. It increases the number of replicas when the number of requests is high. Furthermore, it reduces the number of replicas to save the storage capacity. The lack of considering the number of access time and number of recently used is the main disadvantage of this method.

2. The Dynamic Priority-based Replication Mechanism

In a data grid, when a data file is requested by a user, but is not available on the local storage, it should be replicated or read remotely. If the file is replicated in the local storage, it can be again accessed quickly in the future. This process leads the job execution time to be reduced considerably. Whereas file storages have limited capacity, an appropriate replication strategy should be used on grid networks to enhance the network efficiency. The replication process can be accurate on data grids by using a dynamic priority-based mechanism. A weighted priority-based replication scheme, called WPRS, is proposed in this paper to carry out the file replication process on data grids. It specifies a distinct value for each file exists in the local storage. This value is calculated based on three parameters including file price, number of access time, and present time of the file in the storage. When a file is requested by a user, but does not exist in the local storage, it is hired from other sites. If the local storage does not have enough space to store the hired file, the file having the least value will be removed from the storage using the WPRS mechanism.

2.1 The System Model

A schema of the considered system model is shown in Figure 1. There is a main scheduler to manage the whole network and there are four sites to accomplish the users' requests at anywhere. Each site involves a local scheduler, site information, a local storage, and a desired number of users. The local scheduler manages all activities of the site and communicates with the main scheduler about the required files. The site information includes various features of the site such as number of files, number of users, and site location. The local storage contains some of the data files which are used to accomplish the users' requests. The main scheduler and local schedulers control the file replication process among various sites. When a file does not exist in a site, it is hired from other sites. If there is not enough space in the local storage to store the new hired file, one or more files are removed by the proposed replication mechanism to free up the local storage.

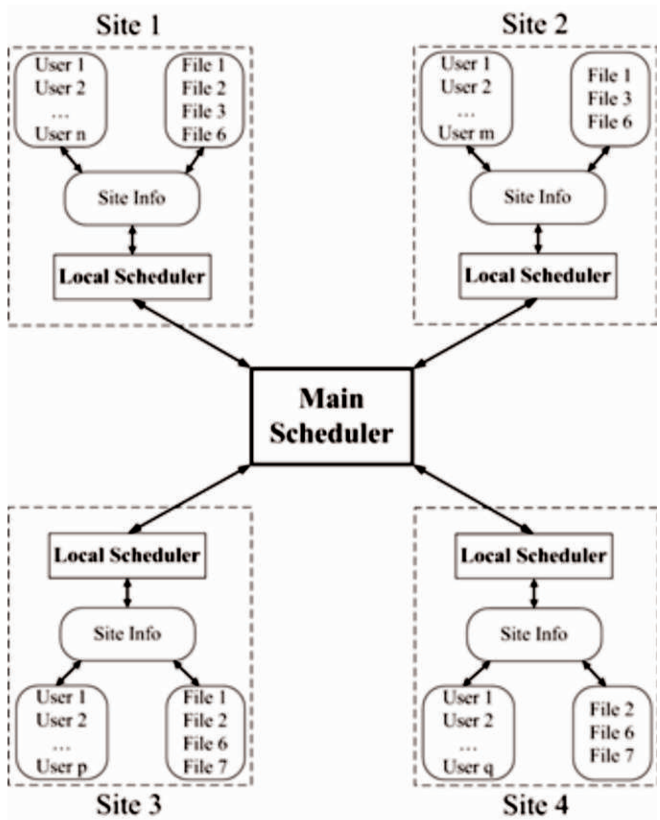


Figure 1. A Schematic of the System Model

2.2 Details of the Proposed Replication Mechanism

The work flow of the proposed replication mechanism is illustrated in Figure 2. Each user can request various data files via sending a message to the local scheduler. If the requested file exists in the local storage, it will be available to the user during execution time; otherwise, the file will be searched on the whole network by the main scheduler. If the file does not exist in one of the sites, the local scheduler alerts the user about the missing file; otherwise, the replication process will be performed by the main scheduler and the local scheduler. In the replication

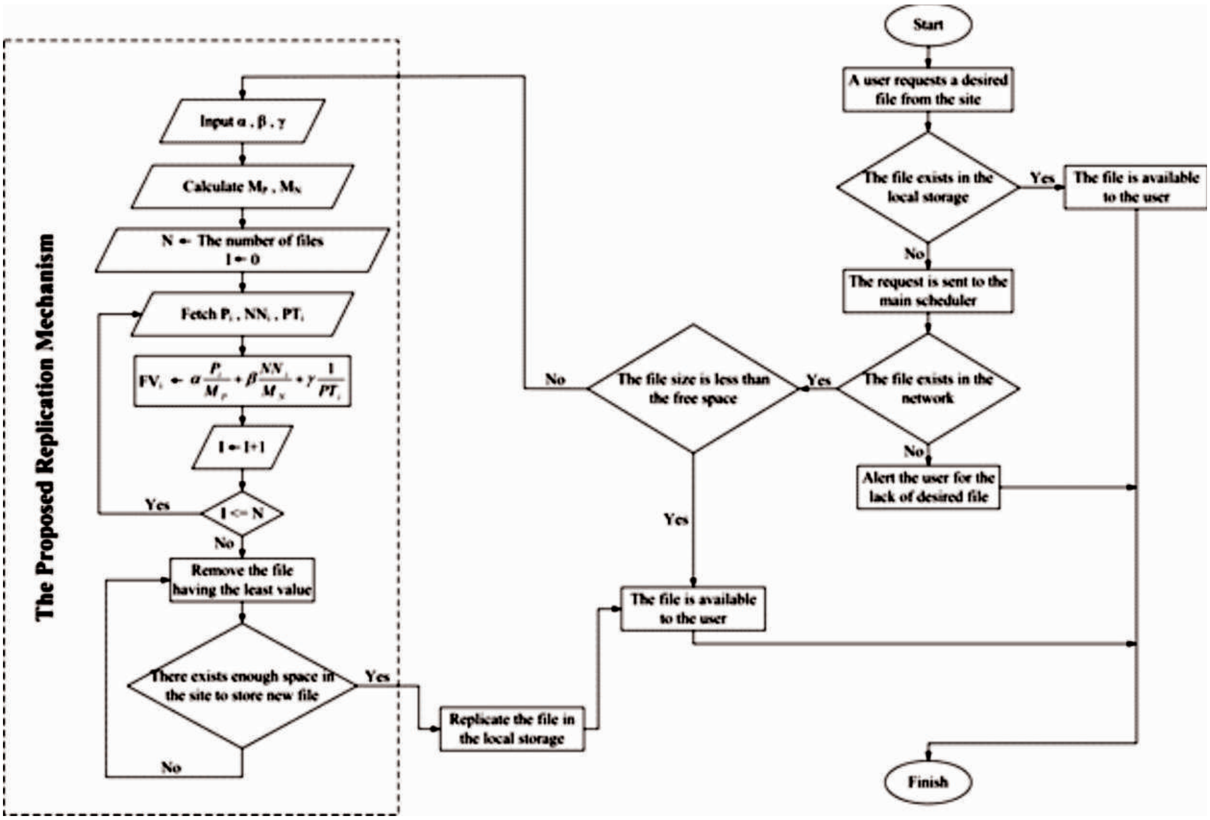


Figure 2. The Flowchart of the Proposed Replication Mechanism

procedure, the requested file is hired from other sites to accomplish the user's request. Since a local storage has a limited capacity, the available storage shall be determined in order to assure the copy process. If the file fits in the available local storage, the file will be copied to the storage; otherwise, one or more files are removed from the storage while free space of the storage becomes more than the file size. The proposed mechanism specifies a value for each file exists in the local storage based on a weighted scheme. The file having the least value will be removed from the storage. The file value is calculated as,

$$FV_i = \alpha \frac{P_i}{M_P} + \beta \frac{NN_i}{M_N} + \gamma \frac{1}{PT_i} \qquad (1)$$

where i denotes the identifier of each stored file, α is the effect of the file price, β is the effect of the number of access time, γ is the effect of the present time, P_i represents the price of file i, M_P represents the maximum price of stored files, NN_i indicates the number of access time to file i, M_N indicates the maximum number of access time, and PT_i indicates the present time of file i in the storage. The above equation represents that price and number of access time have positive impacts as well as present time has negative impact on the file value. That is, if price and number of access time are high as well as present time is low, file value will be high. Note that the file price and the maximum price of stored files are determined by the main scheduler. The present time of each file is calculated as,

$$PT = T_C - T_I \qquad (2)$$

where T_C is the current time and T_I is the time which the file exists in the storage or is copied from other sites.

The parameter M_N is calculated by the main scheduler according to the procedure represented in Algorithm 1. Lines 1 and 2 indicate the input parameters of the algorithm. Two variables are initialized in lines 3 and 4 in order to maintain the cycle number and the maximum price of stored files, respectively. The maximum price is selected from line 5 to line 10. Finally, it is returned to main program in line 11.

Algorithm 1. Calculate the parameter MN by the main scheduler

N ←number of stored files

NA[] ←number of access time

I ←1

Max ←0

while (I ≤N)

if (NA[I] > Max)

Max ←NA[I]

end if

I ←I + 1

end while

return Max

The file having the least value is selected to be removed from a local storage according to the procedure explained in Algorithm 2. The input parameters of the algorithm are defined in lines 1 and 2. Three variables are initialized from line 3 to line 5 in order to maintain the cycle number, the least file value, and the file identifier having the least value. The minimum value and the file identifier including the least value are selected from line 6 to line 12. Finally, the file identifier is returned in line 13.

Algorithm 2. Select the file having the least value to remove from a local storage

N ←number of stored files

V[] ←values of the stored files

I ←2

Min ←V[1]

Index ←1

while (I ≤N)

if (V[I] < Min)

Min ←V[I]

Index ←I

end if

I ←I + 1

end while

return Index

As illustrated in Figure 1, the proposed replication method is applicable for one-level data grid systems. That is, this method is appropriate for the data grids that include a main scheduler and several local schedulers on different areas. Therefore, it cannot be used for multi-level grid systems that have many clusters and servers on a large area.

3. Performance Evaluation

To demonstrate the efficiency of the proposed replication mechanism, it is compared to the LRU and LFU methods in terms of the number of successful jobs, number of non-successful jobs, and buy price. Furthermore, the effect of the simulation time, file size, and number of files on the above parameters is investigated in the analysis results. Note that LRU and LFU are two popular replication methods in data grids. All of the simulated methods use the same simulation conditions, such as simulation time, number of sites, and number of users.

3.1 Simulation Setup

The simulations of the three replication methods are carried out in 5000 rounds. The number of users is randomly determined for each site in the range of [5, 50]. For the whole network, the number of files is randomly specified in the range of [10, 40], the file size is randomly specified in the range of [100, 500], and the file price is randomly determined in the range of [20, 200]. A parameter, called file selection probability is used in the deployment phase to determine the probability degree of each file to being participated in the local storages. The users' requests are generated by the uniform distribution (Granville & Rudnick, 2007; Kuipers & Niederreiter, 2012) that are equal to 0.01 request/round. Simulation parameters are represented in Table 1.

The number of successful jobs equals the number of requested files already locally stored. The number of non-successful jobs is determined as,

$$N_N = N_R - N_S \quad (3)$$

where N_R is the number of generated requests along the simulation process and N_S is the number of successful jobs. Furthermore, the buy price obtained by the

Parameter	Default Value
Simulation time (rounds)	5000
The number of sites	4
The minimum number of users for each site	5
The maximum number of users for each site	50
Minimum number of files	10
Maximum number of files	40
Minimum file size	100 MB
Maximum file size	500 MB
Minimum file price	20 $
Maximum file price	200 $
File selection probability	0.3
Request generation rate	0.01 request/round
α	0.3
β	0.4
γ	0.3

Table 1. Simulation Parameters in the Performance Evaluation Process

simulated methods is calculated as,

$$P = N_N \times F_P \quad (4)$$

where N_N is the number of non-successful jobs and F_P is the price of the file which is hired from other sites.

3.2 Comparison Results

The impact rate of the simulation time on the performance of LRU, LFU, and WPRS methods is shown in Figure 3. Simulation results demonstrate that the proposed replication mechanism has a high efficiency compared to other methods in terms of the number of successful jobs, number of non-successful jobs, and buy price. Figure 3(a) illustrates that the number of successful jobs obtained by the WPRS mechanism is a bit more than that obtained by the LRU and LFU methods. Figure 3(b) shows that the number of non-successful jobs in the proposed replication mechanism is less than other methods along the simulation time. As shown in Figure 3(c), total buy prices of the hired files in the LRU and LFU methods are more than the prices in the proposed mechanism under various simulation times. Note that the quantitative amounts of all simulation results are increased while increasing the simulation time. The reason is that if the simulation time is increased then the number of generated requests will be high; thereby, the number of successful jobs, number of non-successful jobs, and buy price will be increased considerably.

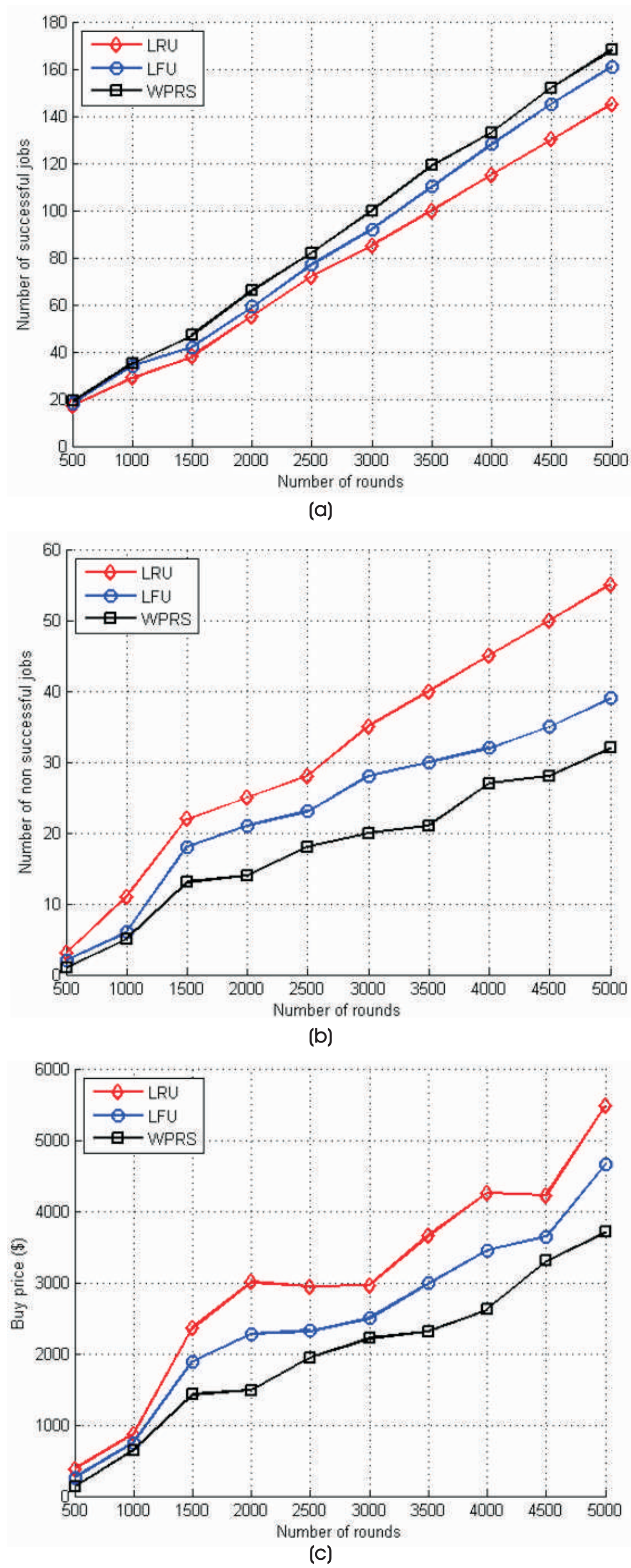
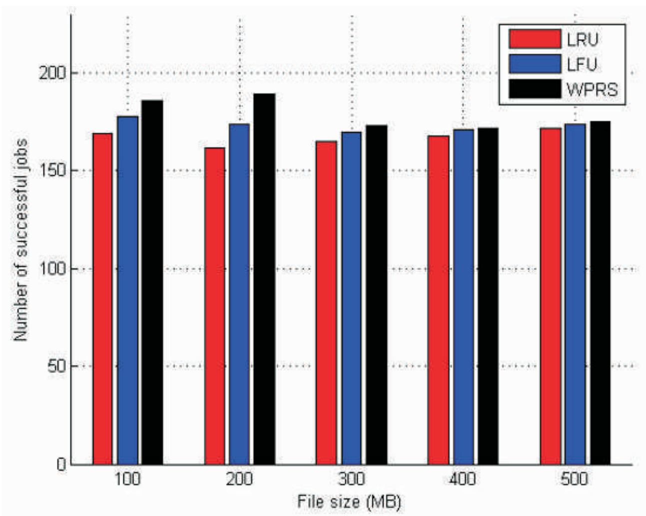
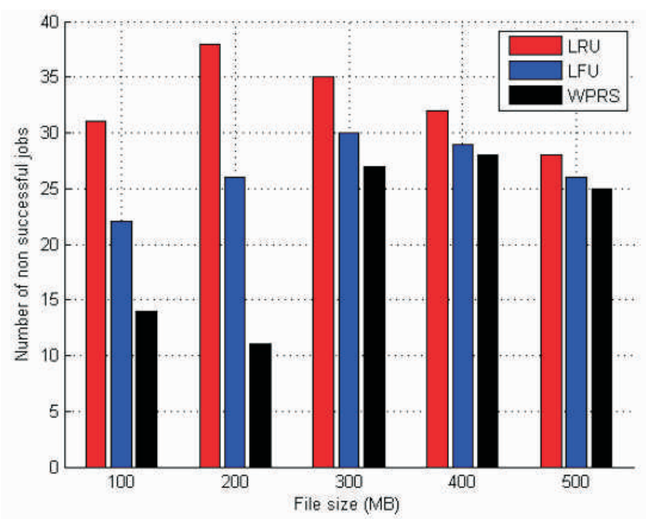


Figure 3. Effect of the Simulation Time on some Dependent Parameters (a) Number of Successful Jobs; (b) Number of Non-Successful Jobs; (c) Buy Price

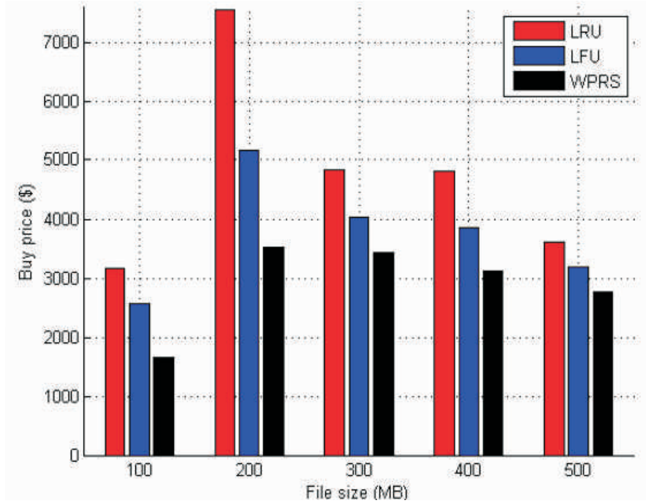
The impact rate of the maximum file size on the efficiency of the LRU, LFU, and WPRS methods is shown in Figure 4. The number of successful jobs, number of non-successful jobs, and buy price obtained by the three replication methods are compared to each other under various file sizes. Figure 4(a) shows that the number of successful jobs in the proposed replication mechanism is high compared to other simulated methods. However, the difference between the performances of the methods is near to each other under various changes on file sizes. Figure 4(b) indicates that the number of non-successful jobs achieved by the proposed mechanism is less than that achieved by the other methods. As illustrated in Figure 4(c), the buy prices of hired files in the WPRS mechanism is less than the LRU and LFU methods. Since



(a)



(b)



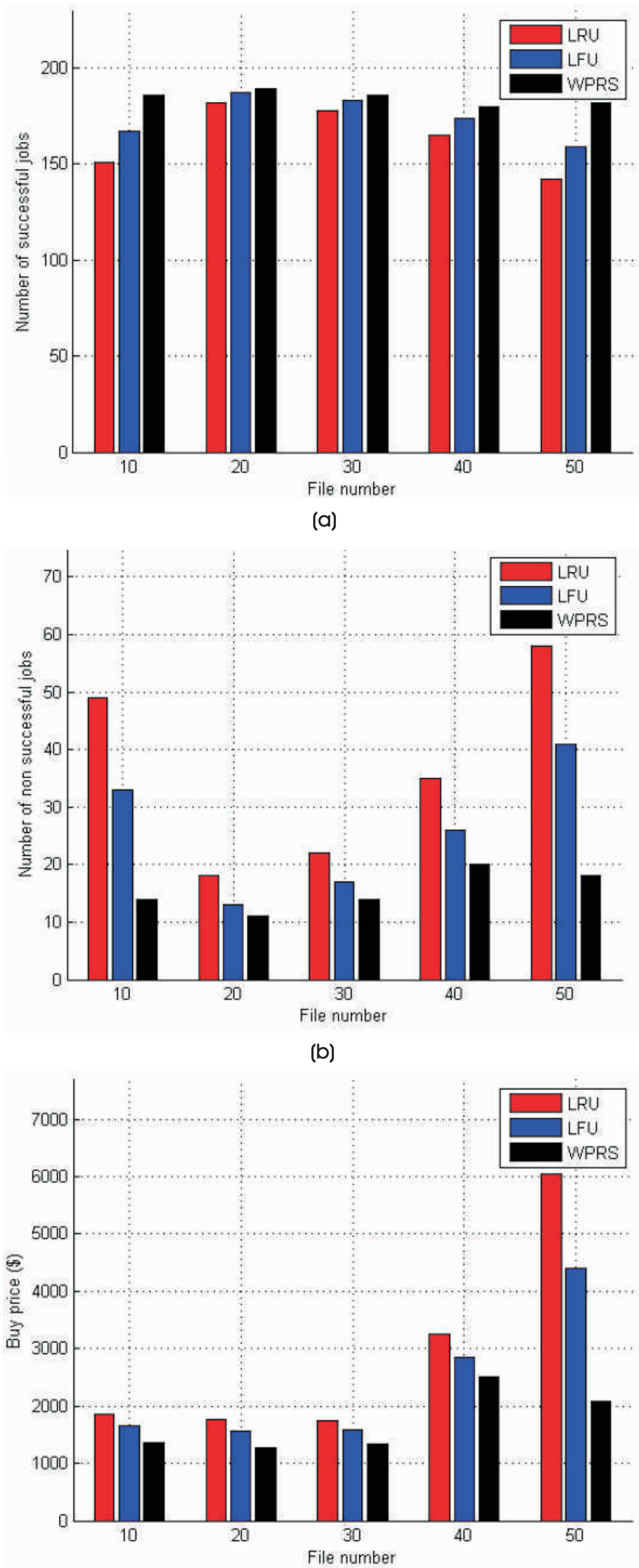
(c)

Figure 4. Effect of the File Size on some Dependent Parameters (a) Number of Successful Jobs; (b) Number of Non-Successful Jobs; (c) Buy Price

the file size does not directly affect on the request generation rate, the simulation results do not include any ascending or descending conditions.

The impact rate of the maximum number of files on the functionality of simulated replication methods is depicted in Figure 5. As shown in Figure 5(a), the number of successful jobs has been constantly higher using the proposed mechanism than that using other replication methods. The reason is that the proposed replication mechanism carefully anticipates the future conditions of network in the replication process in order to decrease the number of missed files in the local storages. Figure 5(b) illustrates that the number of non-successful jobs in the WPRS mechanism is considerably less than the LRU and LFU methods. It demonstrates that the efficiency of the proposed replication mechanism is higher than the other methods in terms of the number of non-successful jobs. Figure 5(c) shows that the buy price achieved by the proposed mechanism is less than that achieved by the other replication methods. When the number of non-successful jobs increases, the buy prices of hired files will be enhanced too. Note that the number of files does not affect on the request generation rate; thereby, the simulation results do not involve any ascending or descending situations.

As demonstrated in the comparison results, the weighted



(a)

(b)

(c)

Figure 5. Effect of the Number of Files on some Dependent Parameters (a) Number of Successful Jobs; (b) Number of Non-successful Jobs; (c) Buy Price

replication procedure used in the proposed WPRS mechanism works efficiently. Therefore, the proposed replication leads the number of successful jobs to be increased, the number of non-successful jobs to be decreased, and the buy prices of hired files to be reduced under various changes on the simulation time, file size, and number of files.

Conclusions

The replication process is used in data grids when a resource (e.g., a data file) is requested by a user, but does not exist in the local storage. An appropriate replication strategy can increase the efficiency of a data grid, considerably. In this paper, a weighted priority-based replication scheme, called WPRS has been proposed for replication procedure in a data grid. If a requested file is not available on the site, it is hired from other sites. If there is not enough space in the local storage to store the hired file, one or more existing files should be removed from the storage. The proposed mechanism carries out this process via specifying a distinct value for each file exists in the local storage. The file value is calculated based on the file price, number of access time, and present time of the files on local storages. The files having the least values are removed from the local storage while the free space of the storage has become less than the size of the requested file. The replication process is performed by a local scheduler placed at each site and a main scheduler. The main scheduler and local schedulers manage all replication procedures throughout the network to increase the efficiency of the proposed replication mechanism.

The proposed replication mechanism is compared to Least Recently Used (LRU) and Least Frequently Used (LFU) methods under various simulation parameters. Simulation results show that the WPRS mechanism has a high efficiency compared to the LRU and LFU methods in terms of the number of successful jobs, number of non-successful jobs, and buy prices of the hired files. They demonstrate that the number of successful jobs obtained by the WPRS mechanism is increased by nearly 15% more than that obtained by the LRU method and by nearly 10% more than that obtained by the LFU method. The number of non-successful jobs achieved by the WPRS mechanism is decreased by nearly 55% less than that achieved by the LRU method and by nearly 40% less than that achieved by the LFU method. Furthermore, the buy price obtained by the WPRS mechanism is reduced by nearly 45% less than that obtained by the LRU method and by nearly 30% less than that obtained by the LFU method.

In the future work, some of the intelligent techniques (e.g., fuzzy logic and neural networks) will be used in the replication mechanisms. This feature could lead to enhance the efficiency of replication process and thereby, increase the number of successful jobs, decrease the number of non-successful jobs, and reduce buy price. Moreover, the scope of further research would be studied on multi-level grid systems (e.g., Amazon Web Services) that contain many clusters and servers throughout the world. It leads the new method would be applicable for the most existing data grid systems.

References

[1]. **Al Mistarihi, H. H. E., & Yong, C. H. (2008).** Replica management in data grid. *International Journal of Computer Science and Network Security,* 8(6), 22-32.

[2]. **Alkhanak, E. N., Lee, S. P., Rezaei, R., & Parizi, R. M. (2016).** Cost optimization approaches for scientific workflow scheduling in cloud and grid computing: A review, classifications, and open issues. *Journal of Systems and Software,* 113, 1-26.

[3]. **Chamola, V., Sikdar, B., & Krishnamachari, B. (2017).** Delay aware resource management for grid energy savings in green cellular base stations with hybrid power supplies. *IEEE Transactions on Communications,* 65(3), 1092-1104.

[4]. **Chang, R. S., Chang, H. P., & Wang, Y. T. (2008, March).** A dynamic weighted data replication strategy in data grids. In *2008 IEEE/ACS International Conference on Computer Systems and Applications* (pp. 414-421). IEEE.

[5]. **Gharajeh, M. S. (2015).** The significant concepts of cloud computing: Technology, architecture, applications, and security. *CreateSpace Independent Publishing Platform.*

[6]. **Gharajeh, M. S. (2017).** Applications of virtualization technology in grid systems and cloud servers. In P. K. Das and G. C. Deka (Eds.), *Design and Use of Virtualization Technology in Cloud Computing* (pp. 1-28). Hershey, PA: IGI Global

[7]. **Gill, S. S., Chana, I., Singh, M., & Buyya, R. (2018).** CHOPPER: An intelligent QoS-aware autonomic resource management approach for cloud computing. *Cluster Computing,* 21(2), 1203-1241.

[8]. **Granville, A., & Rudnick, Z. (Eds.). (2007).** Equidistribution in number theory, an introduction (Vol. 237). *Springer Science & Business Media.*

[9]. **Hossain, E., Han, Z., & Poor, H. V. (Eds.). (2012).** *Smart Grid Communications and Networking.* Cambridge University Press.

[10]. **Jinno, M., Kozicki, B., Takara, H., Watanabe, A., Sone, Y., Tanaka, T., & Hirano, A. (2010).** Distance-adaptive spectrum resource allocation in spectrum-sliced elastic optical path network [topics in optical communications]. *IEEE Communications Magazine,* 48(8), 138-145.

[11]. **Krašovec, B., & Filipčič, A. (2019).** Enhancing the Grid with Cloud Computing. *Journal of Grid Computing,* 17(1), 119-135.

[12]. **Kuipers, L., & Niederreiter, H. (2012).** *Uniform Distribution of Sequences.* Courier Corporation.

[13]. **Mahmood, A., Ullah, M. N., Razzaq, S., Basit, A., Mustafa, U., Naeem, M., & Javaid, N. (2014).** A new scheme for demand side management in future smart grid networks. *Procedia Computer Science,* 32, 477-484.

[14]. **Mansouri, N., & Dastghaibyfard, G. H. (2013).** Enhanced dynamic hierarchical replication and weighted scheduling strategy in data grid. *Journal of Parallel and Distributed Computing,* 73(4), 534-543.

[15]. **Mariotti, M., Gervasi, O., Vella, F., Cuzzocrea, A., & Costantini, A. (2018).** Strategies and systems towards grids and clouds integration: A DBMS-based solution. *Future Generation Computer Systems,* 88, 718-729.

[16]. **Mishra, M. K., Patel, Y. S., Ghosh, M., & Mund, G. B. (2017).** A review and classification of grid computing systems. *International Journal of Computational*

Intelligence Research, 13(3), 369-402.

[17]. Nazir, B., Ishaq, F., Shamshirband, S., & Chronopoulos, A. (2018). The impact of the implementation cost of replication in data grid job scheduling. *Mathematical and Computational Applications,* 23(2), 1-19.

[18]. Nicholson, C., Cameron, D. G., Doyle, A. T., Millar, A. P., & Stockinger, K. (2008). Dynamic data replication in lcg 2008. *Concurrency and Computation: Practice and Experience,* 20(11), 1259-1271.

[19]. O'neil, E. J., O'neil, P. E., & Weikum, G. (1993). The LRU-K page replacement algorithm for database disk buffering. *ACM Sigmod Record,* 22(2), 297-306.

[20]. Prischepa, V. (2004). An efficient web caching algorithm based on LFU-K replacement policy. In *Proceedings of the Spring Young Researcher's Colloquium on Database and Information Systems* (pp. 1-5). IEEE.

[21]. Rahman, R. M., Barker, K., & Alhajj, R. (2009). Performance evaluation of different replica placement algorithms. *International Journal of Grid and Utility Computing,* 1(2), 121-133.

[22]. Ranganathan, K., Iamnitchi, A., & Foster, I. (2002, May). Improving data availability through dynamic model-driven replication in large peer-to-peer communities. In *2nd IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGRID'02)* (pp. 376-376). IEEE.

[23]. Sashi, K., & Thanamani, A. S. (2011). Dynamic replication in a data grid using a modified BHR region based algorithm. *Future Generation Computer Systems,* 27(2), 202-210.

[24]. Tang, M., Lee, B. S., Tang, X., & Yeo, C. K. (2006). The impact of data replication on job scheduling performance in the Data Grid. *Future Generation Computer Systems,* 22(3), 254-268.

[25]. Wang, L., Jie, W., & Chen, J. (2018). *Grid Computing: Infrastructure, Service, and Applications.* CRC Press.

[26]. Weng, Y., Negi, R., Faloutsos, C., & Ilić, M. D. (2017). Robust data-driven state estimation for smart grid. *IEEE Transactions on Smart Grid,* 8(4), 1956-1967.

[27]. Wilkinson, B. (2009). *Grid Computing: Techniques and Applications.* CRC Press.

ABOUT THE AUTHOR

Mohammad Samadi Gharajeh received ASc in Computer Software from University of Applied Sciences and Technology in 2005, BSc in Engineering of Computer Software Technology from University of Applied Sciences and Technology in 2009, and MSc in Computer Engineering – Computer Systems Architecture from Islamic Azad University in 2013. He has already developed various software systems, simulations, intelligent systems, and research projects. His research interests include Artificial Intelligence and Software Systems, focusing on the Application of Artificial Intelligence Techniques to Real World Problems in Domains such as Robotics, Computer Vision, and Computer Networks. He was a Technical Program Committee Member and a Reviewer in some of the International Conference Proceedings. Besides, he is an Editorial Board member and a Reviewer in some of the International Scientific Journals, a Lecturer of university, and an IEEE Member now.