

A new fuzzy optimal data replication method for data grid

Zeinab Ghilavizadeh^{a*}, Seyed Javad Mirabedini^b and Ali Harounabadi^b

^aDepartment of Computer, Science and Research of Bushehr Branch, Islamic Azad University, Bushehr, Iran

^bDepartment of Computer, Central Tehran Branch, Islamic Azad University, Tehran, Iran

CHRONICLE

Article history:
Received October 1, 2012
Received in revised format
22 January 2013
Accepted 23 January 2013
Available online
January 24 2013

Keywords:

Data Grid
Dynamic data replication
Data replacement
Replica
Fuzzy Logic

ABSTRACT

These days, There are several applications where we face with large data set and it has become an important part of common resources in different scientific areas. In fact, there are many applications where there are literally huge amount of information handled either in terabyte or in petabyte. Many scientists apply huge amount of data distributed geographically around the world through advanced computing systems. The huge volume data and calculations have created new problems in accessing, processing and distribution of data. The challenges of data management infrastructure have become very difficult under a large amount of data, different geographical spaces, and complicated involved calculations. Data Grid is a remedy to all mentioned problems. In this paper, a new method of dynamic optimal data replication in data grid is introduced where it reduces the total job execution time and increases the locality in accessibilities by detecting and impacting the factors influencing the data replication. Proposed method is composed of two main phases. During the first phase is the phase of file application and replication operation. In this phase, we evaluate three factors influencing the data replication and determine whether the requested file can be replicated or it can be used from distance. In the second phase or the replacement phase, the proposed method investigates whether there is enough space in the destination to store the requested file or not. In this phase, the proposed method also chooses a replica with the lowest value for deletion by considering three replica factors to increase the performance of system. The results of simulation also indicate the improved performance of our proposed method compared with other replication methods represented in the simulator Optorsim.

1. Introduction

These days, there are huge amount data and calculations geographically distributed around the world and these have created some challenges of accessing, processing and distributing data. In fact, it is a tedious task to handle large amount of data to do complicated calculations. Data Grid is an alternative solution for these kinds of problems, which is an architecture for management analyses of large

*Corresponding author.
E-mail addresses: z_ghilavi@yahoo.com (Z. Ghilavizadeh)

scientific data sets (Chervenak et al., 2000). Data Grid is a development concept based on “Grid”, which is an infrastructure for integrating distributed computational elements. In an advanced distributed environment of a grade, there are some data access matrices, response time, access cost, bandwidth utilization, reliability, and scalability. Data replication is an alternative technique to use to handle different challenges of high accessibility. In this method, we increase the fault-tolerance and improve the scalability and response time using data high accessibility and low bandwidth consumption (Amjad et al., 2012; Foster, 2002; Ranganathan et al., 2001; Lamahamedi & Szymanski, 2002; Ranganathan et al., 2002; Rahman et al., 2005; Vazhkudai et al., 2001; Stockinger et al., 2002).

When data is replicated, the replicas are built from the data files from various sites in a data grade. Replication can save the storage resources while occupying the provided data storage on each site. It also saves a large amount of bandwidth when the data is presented only on a particular site. Therefore, for providing a fast data access at all the times, data replication is an excellent exchange between the storage accessibility and bandwidth accessibility (Yuan et al., 2007). The main idea is to maintain the data near the user for effective and quick access. The data replication algorithm should respond to the critical questions such as the data type needed for duplication and the place they must be kept. Grid users' dynamic behavior makes it different for making difficult decisions about data replication to achieve the maximum accessibility (Schintke, & Reinefeld, 2003).

This paper introduces a new method of dynamic data replication in the data grid. The introduced method reduces the total job execution time by grid sites by involving the factors influencing the data replication. Therefore, the primary objective is to reduce the data access time, to reduce the data access latency, to reduce the job execution time, to reduce the occupied bandwidth, to increase the locality in accessibilities and to replicate data on the grid optimally.

2. Literature review

Zhao et al. (2010) provided a Dynamic Optimal Replication Strategy (DORS), where they extracted a replacement algorithm of new data replica based on the accessibility record, size of files and the network conditions. Therefore, the replicas with minimum amount are removed and the local node has sufficient capacity to load the new file. This algorithm has been used for dynamic data grid environment and can increase the accessibility output of data replica and there are some strategies used in two stages. In the first step, the replication strategy has been suggested; it makes decision based on the number of file replicas to decide whether the requested file should be kept in local storage system or not. In the second step, a new replacement method for new replication, in which the files are computed in the local storage system, is embedded. The file with high capacity will be used to improve the efficiency of data access, thus they should be protected.

Chaprada et al. (2010) suggested a strategy for dynamic data grids, which helps to increase file accessibility to improve the response time and reduce the bandwidth consumption. This strategy is evolutionary; in other words, it helps to achieve the load sustainability of the whole grid at desired amount of time. The strategy consists of the following steps: Selecting the best candidate files for replication, determining the best sites to locate the files selected in the previous step, choosing the best replica. The first step of the strategy to detect the candidate files for replication is based on the following parameters: The number of times where the file F_j is requested, and the number of available replica of a file F_j on all grid system. The strategy investigates that a file F_j should be replicated if it has been essential and no enough replicas of F_j are available on the grid. After selecting the best candidate files for the replication, the best sites should be determined for their location. In order to be successful in this selection, the strategy takes the following parameters into account: Number of requests for each file F_j with each site S_i , utilization of each site S_i based on the grid when a file F_j is requested to replicate or to satisfy a request of the best replication. Selecting the

best replication not only depends on the bandwidth, but also is dependent on the utilization function of grid sites. In fact, if we want to transfer a file F_K from a site S_j to a site S_i , it is possible to reduce the transferring time, S_j is nearest to S_i . A site, which is always connected to the grid, has no equal utilization with the other sites, which is frequently connected and disconnected. For these reasons, the strategy will utilize jointly these two parameters (bandwidth and utilization).

Abdurrah and Xie (2010) are believed to be the first who introduced FIRE Strategy. Consider a data grid, which consists of n sites demonstrated with set $S = \{s_1, \dots, s_j, \dots, s_n\}$. Moreover, let $F = \{f_1, \dots, f_m, \dots, f_i\}$ be the data grid and let j be required files with $D = \{d_1, \dots, d_k, \dots, d_w\}$ in which $w \leq m$. For each file d_k in D , FIRE checks whether it is locally available or not and, if not, FIRE tests whether local SE has a space to adapt d_k . If yes, FIRE replicates d_k from a site with minimum cost based on auction protocol (AP). If not, FIRE compares the number of remote access on d_k sent from the s_1 against the number of local access of the least accessed file locally from the site s_1 . If the earlier were not lower than the latter, the least local accessed file would be deleted and d_k could be replicated on s_1 by using AP. Otherwise, the work j should have remote access to d_k . In this case, the replication manager of a remote site with d_k file should update his own access table to save this remote access event. Finally, the replication manager of site s_1 updates his/her own file access table to record the local and remote access created by the work j .

3. A new fuzzy optimal data replication method

The primary objective of the proposed method of this paper is to detect important factors, playing effective role in increasing the efficiency of grid, decreasing the response time and reducing the bandwidth occupation, and to implement these factors in the replication algorithm, properly. The important point in the proposed method of this paper is that the effective factors in replication should be found and we should determine the effectiveness of each of these factors. Another important point in this study is the way of using these factors, simultaneously in the replication algorithm or in other words the way of valuating them.

3.1 Architecture of proposed method

Architecture of Fuzzy Optimal Replication Method (FORM) are shown in Fig.1.

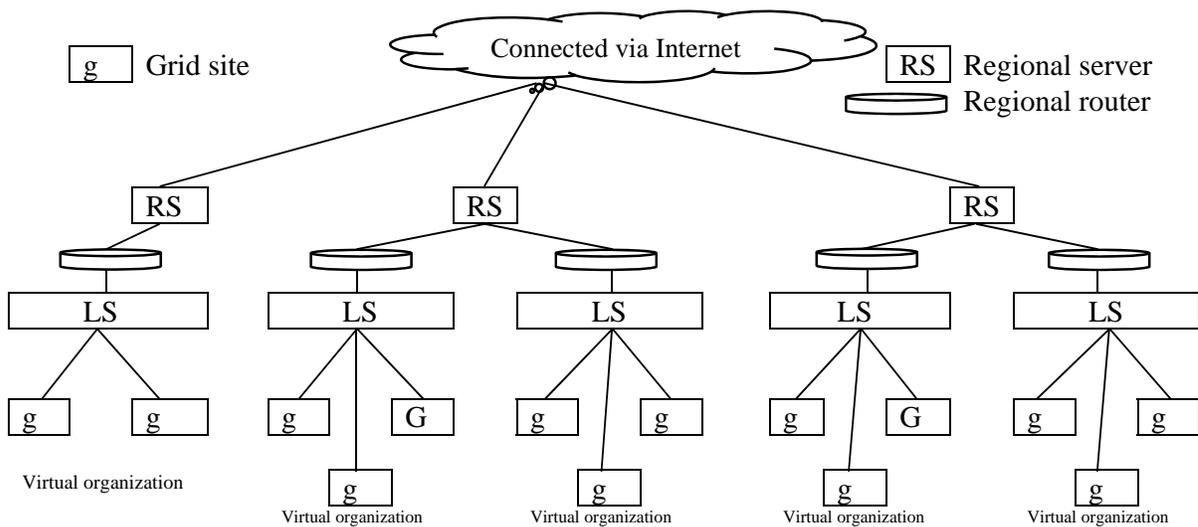


Fig.1. Architecture of FORM (Saadat & Rahmani, 2012).

Grid sites are located at the lowest level of this architecture and these sites have been composed of storage as well as computational elements. A virtual organization is built from the total of multiple sites. There is a local server (LS) for each virtual organization where LS is responsible for maintaining the sequence of access to data inside each virtual organization. Moreover, the list of replicas is also placed in this component. Note that the speed of data access within each virtual organization is higher than the speed of data access on other virtual organizations. Regional Server (RS) is at the higher level of virtual organization, where each RS monitors one or more virtual organizations and they are connected through the Internet. Thus, the data transfer rate among them is less than the data transfer rate within the virtual organizations.

3.2 Description of FORM Strategy

The proposed method of FORM is composed of two phases:

3.2.1 First Phase: The phase for requesting a file and doing the replication

In this phase, first, we consider effective factors on whether the requested file can be replicated or it can be implemented from far distance. These factors include the availability of storage space remained on the local site, the replication expenditure of target file and the number available replicas from the requested file on the total grid. In fact, given that the replication is an expensive action, we need to know whether the replication is cost effective at every stage or not. This investigation has been performed properly in the proposed method of this paper and has been led to some appropriate results in saving the storage time and space and repeating the replication operation.

First factor: The remaining available storage space on the local site. In proposed method of FORM, it is investigated before any replication action whether the local site has sufficient capacity to store the requested file or not. If the answer is yes and there is sufficient capacity on the site, the file replication is executed on the site and replica of files are recorded on the local site. Otherwise, if the local site does not have sufficient space for file storage, we need to decide whether the requested file can be either available from distance for the local site or it needs to be replicated on the local site. This decision-making in our way is executed through evaluating two other factors investigated as follows. The important point is that if the replication is performed blindly without considering the conditions, the files available from distance and with minimum cost are replicated and replaced with important and popular files due to the lack of storage space.

Second factor: The cost of accessing the target file, which plays the key role in each replication algorithm. In fact, the objective of replication is to reduce the job execution time. We the cost of transferring necessary files from the source site to the destination site is reduced, obviously, the time of performing the jobs will be reduced, significantly. In the proposed method of FORM, we investigate whether a file requested by a local site can be implemented from distance in the event there is limited storage space or not. If the local site does not have sufficient space to store the requested file, we should pay the associated replication cost. If the cost of accessing to the requested file is larger than the predetermined threshold amount, it means that the access to the file from remote distance will become time consuming. Moreover, given this point that the sites in a virtual organization have similar interests and also the bandwidth among the sites in a virtual organization is much lower than the bandwidth among other organizations, the proposed method of this paper has adopted this policy, which keeps, at least, a replica of requested file inside it for each virtual organization. As a result, if any other sites in a virtual organization needs to have the access to a file in near future and the file is not available locally, they can be replicated very easily. Given the mentioned cases, if the access expenditure of a file is more than the threshold value, the file should be replicated for local site because it is more likely that that site and other neighboring sites would request the file. Thus, we need to keep it in place nearer than its previous space to the sites of that

virtual organization. However, if the access cost is less than the threshold value, then our algorithm will investigate the third factor for final decision-making of replication.

Third factor: The number of available replicas from the requested file in the whole grid. At this stage, the proposed method investigates how many replicas of requested file exist in the whole grid space and looks for the conditions, where appropriate numbers should be established. This factor is applied in replication decision making because if the number of replicas available on grid sites becomes more than the pre-specified limit, it will lead to the ineffectual occupation of valuable storage space and thus those valuable and popular files will have no sufficient space to be replicated and inevitably the replication will be used, frequently. Besides, if the number of replica in a file is less than the required limit, it will lead to an increase on the cost of replication. For example, if there is no replica of a file available in a virtual organization or even in the neighboring and near virtual organizations, the access to the closest replica will be very costly and time-consuming for the sites of that virtual organization. If the number of available replicas of requested file is more than the threshold, there is no need to replicate the file and the local site can use the file from remote distance. Otherwise, if the number of available replicas of requested file is less than the threshold value, the local site should replicate the requested file from the closest available location. Therefore, the replication phase ends for a local site and its required file. Replication algorithm of FORM is shown in Fig.2.

```

Replication Algorithm (grid site 'g', file 'f')
1. {
2. If (there is requested file in local site 'g')
3. Do nothing ;
4. else
5. {
6. If (local site 'g' has enough free space)
7. Replicate the best replica of file 'f' from closest storage element to
8. local site 'g' ;
9. else
10. {
11. If (available space on local site 'g' = 0 and
12. Access cost of requested file 'f' < threshold1 and
13. The number of replicas of file 'f' > threshold2 )
14. Local site 'g' must be use requested file 'f' remotely from closest storage element ;
15. } //end of third else
16. else
17. local server decide based on fuzzy algorithm to delete some replicas on local site 'g' for make space for requested file 'f' ;
18. } //end of second else
19. } //end of algorithm

```

Fig. 2. Replication Algorithm of FORM

3.2.2 Second Phase, Replacement Phase

The second phase focuses on subject of replicas replacement. The file, which has less possibility to be requested in the future and if it is requested, the replication cost for that site will not be significant and its replicas are sufficiently available on the site should be removed. Three factors are considered for investigating the removal or not-removal of a replica in the method FORM. These three factors determine the value of replica. If a replica has lower value, it will be the first candidate for replacement. These three factors are as follows:

The number of access to the replica in the past: If the number of access to a replica has been greater than in the past, it is more likely that it will be used in the future. Thus, the value of that replica is higher and it is not a good candidate for deletion.

The cost of replication: An expensive replica is not an appropriate option for deletion because if it is needed in the future, the high cost should be paid for its replication. Thus, the more the cost of replication is, the more the value of replication will be.

Number of available replicas of file in other Grid sites: The less the number of available replica from a file in other Grid sites is, the more the value of that file is and it should be preserved at the local site. The file with numerous replicas throughout the grid is more likely to be deleted from the local site. In addition, the file with few replicas in the whole grid is not a good option to be deleted because if this file is needed in the near future by the same site or neighboring sites, probably it should be replicated from a remote site with the high cost.

Fuzzy logic is used in order to calculate the value of each replica.

3.2.2.1 Evaluating the value of each replica by using the fuzzy logic

Fuzzy logic is used in order to calculate the value of each replica. The Fuzzy Inference System (FIS) is implemented in calculating the value of each replica because the factors building the value of each replica have the uncertainty and ambiguity. MATLAB Fuzzy Logic Toolbox has been used in order to implement the fuzzy method. MATLAB makes it possible for us to create and edit Fuzzy Inference Systems. Our fuzzy inference system consists of three inputs and an output. Input parameters are: 1. Number of access to the replica, 2. Replication Cost, and 3. Number of Available Replicas for a file. Fuzzy output parameter is also the value of that replica.

3.2.2.2 Replacement algorithm of Replicas in the FORM method

```

Replacement Algorithm (Grid Site g, File f)
1. {
2.  if (g.storageSize < f.size)
3.    Access file 'f' remotely, or replicate 'f' to the closest storage element to 'g' ;
4.  else
5.    {
6.    if (g.availableStorageSize < f.size)
7.      replicate file 'f' to grid site 'g' and exit ;
8.    else
9.      {
10.     "selected replicas" = all replicas stored at 'g' ;
11.     for each replica in "selected replicas"
12.       {
13.         RV(replica r) = call Fuzzy Function (Number of Accesses(r), Replication Cost(r) );
14.         Replication Cost(replica r) =  $\frac{size_r}{bandwidth_{ag}}$  + propagation delay time ;
15.       } //end of for each
16.     "selected replicas" = sort the replicas of "selected replicas" in ascending order, according to their RV ;
17.     while ("selected replicas" != empty)
18.       {
19.         select a replica from top of the "selected replicas" and delete it from grid site 'g' ;
20.         if (g.availableStorageSize < f.size)
21.           replicate file 'f' to grid site 'g' and exit;
22.       } //end of while
23.     go to line 11;
24.   } //end of second else
25. } //end of first else
26. } //end of algorithm

```

Fig. 3. Replacement algorithm of FORM

Replicas Replacement algorithm of FORM is shown in Fig.3 (Destination site g, File f).

- Number of Accesses_(r): Number access to the replica r in the past,
- size_r: Size of replica r,

- Replication Cost_(r): Cost of performing the replication for replica r;
- Replica- Numbers _(r): Number of available replicas of file r on other sites,
- RV: Replica Value.
- bandwidth_{ag}: Bandwidth between the sites g and site a. Site a is a site containing the replica r and site g is the destination site (requester),
- Propagation delay time_{ag}: Delay time for sending the replica from the source site (a) to the destination site (g),

4. Simulation and its results

The simulator package "OptorSim" has been used for simulation and evaluating the efficiency of proposed method in this paper (OptorSim – A Replica Optimiser Simulation). OptorSim (Bell et al, 2002) is a simulator package written in Java. The package has been developed for investigating the impact of replica optimizing algorithms in the data grid environment (EDG – The European Data Grid Project) and for demonstrating the real structure of European Data Grid. In this simulator, it is assumed that the grid is composed of multiple sites, each consists of zero or the large number of computational elements and zero or the large number of storage elements. Computational elements execute jobs by processing the files and these files are stored in the storage elements. Server controls the resource of job scheduling operations and allocating them to the computational resources according to the scheduling algorithm. Each site manages the content of its own files by using the Replica Manager (RM). At the heart of the replica administrator, there is the Replica Optimizer (RO), which includes the replication algorithm that is responsible for automatic creation and deletion of replicas. The Algorithm FORM has been compared with five algorithms of No Replication, LRU, LFU, EcoModel, and EcoModel Zipf-Like distribution. The CMS Data Challenge 2002 is the architecture and topology implemented in the simulation. This architecture is shown in the Fig.4 (Cameron et al., 2004). General parameters of simulation are presented in Table 1 and it should be noted that in this simulation the data are read-only.

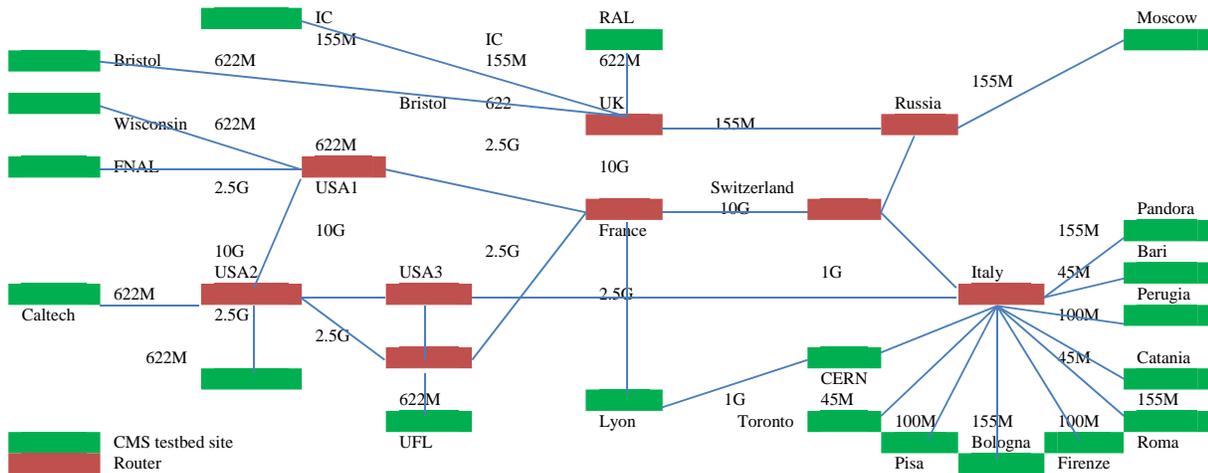


Fig. 4. Architecture used in the simulation of The CMS Data Challenge (2002)

Table1
General Parameters of Simulation

Parameter	Value	Parameter	Value
Total number of sites	20	File access patterns	Sequential, random, random walk unitary, random walk Gaussian, random Zipf access
Number of computational resources - CE	18	Size of each file (GB)	1
Number of storage resources - SE	20	Total size of files (GB)	97
Number of Routers	8	Duration of past access (Milliseconds)	106
Storage capacity of each site (GB)	100, 50	Minimum bandwidth between two sites (Mbps)	45
Number of jobs	100	Maximum bandwidth between two sites (Mbps)	10000
Number of job types	6	Number of repeating the simulation	10
Job delay (milliseconds)	2500		

4.1 Criteria of performance evaluation

4.1.1 Mean job time

The proposed method of FORM has been compared with 5 other algorithms in terms of mean job time. The results of comparing the job time for 6 algorithms in different access patterns have been shown in Fig. 5.

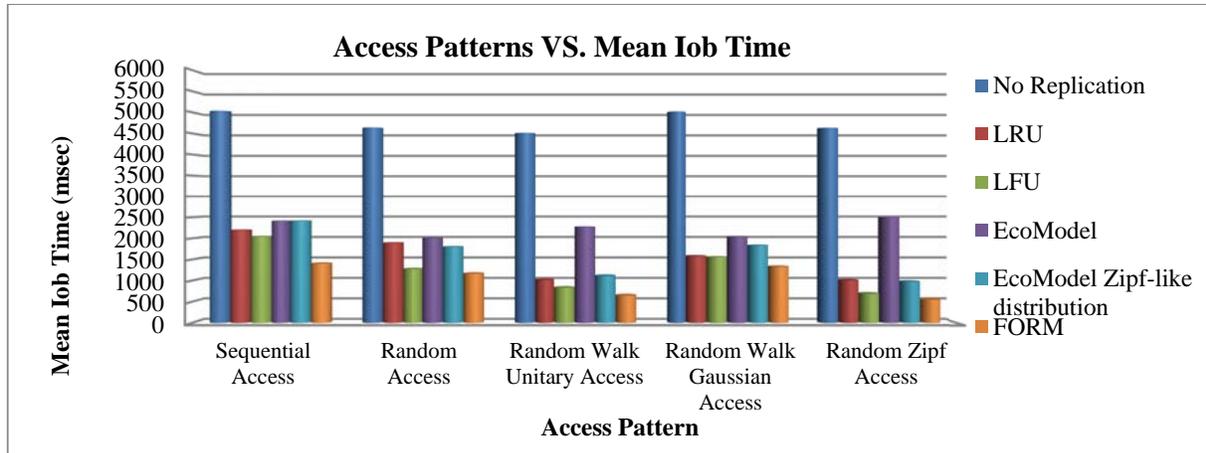


Fig. 5. Diagram of access patterns versus Mean job time

The results of simulation indicate that the proposed method maintains the minimum job execution time compared with other methods and all access patterns. The jobs are executed in less amount of time since the method, FORM, performs the replication wisely and accurately. It involves three important factors of free storage space remained on the local site, the replication cost of target file, and the number of available replica from the requested file in decision-making. It also uses the target file from the remote distance in the case of being cost effective so that the space was remained for more important and widely used files. On the other hand, the algorithm of FORM has determined the value of each file by taking into account three mentioned parameters. It also removes the less valuable files if there is no enough space in the destination to store the new replica. As a result, the valuable files are still remained on Grid sites and are available locally if necessary.

4.1.2 Effective network usage

This criterion is the ratio of transferred files to the requested files. In other words, ENU determines the ratio of replicated files to the local access. Thus if this ratio is smaller it means the used optimal algorithm has worked in the much correct, well, and proper place due to the replication of replicas. Effective network usage has a value between 0 and 1 and is calculated by using the equation 1.

$$ENU = \frac{N_{remote\ file\ accesses} + N_{file\ replications}}{N_{remote\ file\ accesses} + N_{local\ file\ accesses}} \quad (1)$$

The proposed algorithm of FORM was compared with 5 methods in terms of effective network usage. The comparison results are shown in Fig. 6. As shown in these figures, FORM has the lowest value of ENU compared with other algorithms in all access patterns because this algorithm performs these two operations optimally by taking into account the effective factors and applying them correctly in replication and replacement phases. Thus, the number of local access is increased and the number of replication reduced and therefore the ratio of ENU and the amount of occupied bandwidth is decreased.

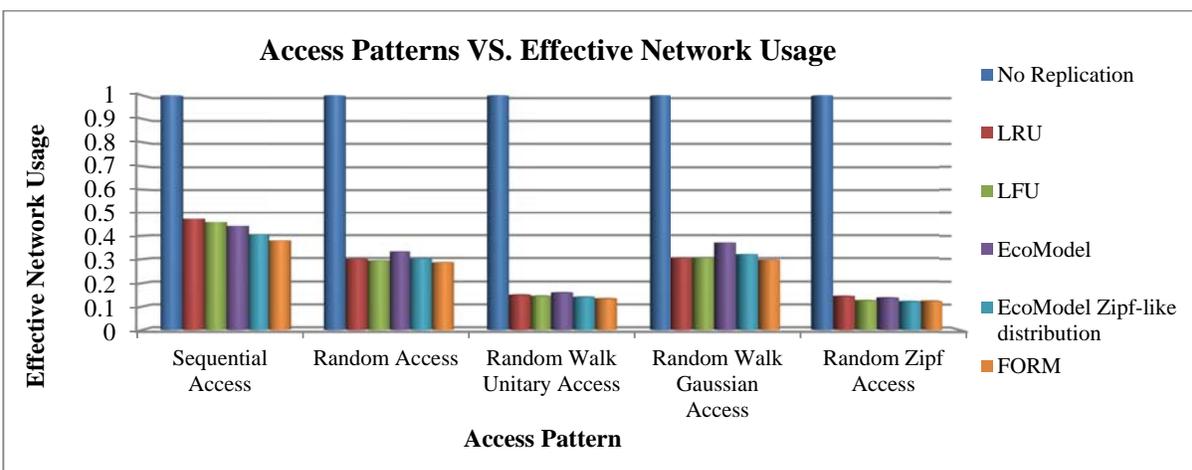


Fig. 6. Diagram of access patterns versus the effective network usage

5 . Conclusion and future works

In this study, a new dynamic data replication method was proposed in data grid, called FORM. The proposed model of this paper aimed to involve different factors, which played the most effective role in increasing the efficiency of grid, decreasing the response time, and reducing the bandwidth occupying, in the replication. Furthermore, the effectiveness by each of these factors was determined in our method. Then the way of simultaneous use of these factors in the replication algorithms or in other words, the way of valuating them were discovered.

The proposed method of FORM had two phases. The first phase was associated with file request and the doing the replication. In this phase, by considering three effective factors (the available storage space in the local site, the replication cost of target file, and the number of available replicas of requested files in the whole grade), it was decided whether the requested file could be replicated or be used from the remote distance. The second phase focused on the replication of replicas. According to the FORM method, three factors including the number of access to the replica in the past, the replication cost, and the number of available replicas of files in other Grid sites were considered for investigating whether we should delete a replica or not and these three values determined the value of replica. If the value of replica is lower, it will be the first candidate for replacement. Fuzzy logic was used in order to calculate the value of each replica. Our proposed algorithm was evaluated and compared along with the existing 5 algorithms in simulator Optorsim according to two efficiency criteria of Mean job time and Effective network usage, and the results of these comparisons expressed this fact that the proposed method of FORM had the better performance Compared to other methods.

- Given the improvements in the proposed algorithm of FORM, it is predicted that the better results are obtained by increasing the number of jobs in the simulation and the better results are obtained for a large number of jobs than small number of jobs. As the future work, we are seeking to achieve this goal by changing the parameter of simulation in the simulator Optorsim.
- Moreover, another restriction on Optorsim is that the sizes of all files in the simulation are considered similar. We are seeking to consider the file sizes different for making the simulation of our proposed method similar to the real environment and change the job configuration file in the stimulator Optorsim.

References

- Abdurrab, A.R., & Xie, T. (2010). *FIRE: A File Reunion Based Data Replication Strategy for Data Grid*. 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing, 1-9.
- Amjad, T., Sher, M., & Daud, A. (2012). *A survey of dynamic replication strategies for improving data availability in data grids*. *Future Generation Computer Systems*, 28, 337–349.
- Bell, W.H., Cameron, D.G., Capozza, L., Millar, A.P., Stockinger, K., & Zini, F. (2002). *Simulation of Dynamic Grid Replication Strategies in OptorSim*. *International Journal of High performance Computing Applications*, 46-57.
- Cameron, D.G., Millar, A.P., Nicholson, C., Schiaffino, R., Zini, F., & Stockinger, K. (2003). *OptorSim: a simulation tool for scheduling and replica optimization in data grids*. *International Journal of High Performance Computing Applications*, 17, 403-416.
- Chaprada, F.B., Ounelli, H., & Chettaoui, H. (2010). *An Efficient Replication Strategy for Dynamic Data Grids*. 1st International Conference on P2P, Parallel, Grid, Cloud and Internet Computing, 50-54.
- Chervenak, A., Foster, I., Kesselman, C., Salisbury, C., & Tuecke, S. (2000). *The data grid: Towards an architecture for the distributed management and analysis of large scientific datasets*. *Journal of Network and Computer Applications*, 23 (3), 187–200.
- EDG – The European DataGrid Project, <http://www.edg.org>.
- Foster, I. (2002). *The grid: A new infrastructure for 21st century science*. *Physics Today*, 55(2), 42-47.
- Lamehamedi, H., & Szymanski, B. (2002). *Data replication strategies in grid environments*, 5th International Conference on Algorithms and Architectures for Parallel Processing, ICA3PP, 378.
- OptorSim – A Replica Optimiser Simulation, <http://grid-data-management.web.cern.ch/grid-data-management/optimization/optor>.
- Rahman, R.M., Barker K., & Alhajj, R. (2005). *Replica placement in data grid: Considering utility and risk*. *IEEE International Conference on Information Technology*, 1, 354-359.
- Ranganathan, K., & Foster, I. (2001). *Design and evaluation of dynamic replication strategies for a high performance data grid*. 1st International Conference on Computing in High Energy and Nuclear Physics, 45-51.
- Ranganathan, K., Iamnitchi, A., & Foster, I. (2002). *Improving data availability through dynamic model-driven replication in large peer-to-peer communities*. *IEEE/ACM 2TH International Symposium on Cluster Computing and the Grid*, CCGrid, 376.
- Saadat, N., & Rahmani, A.M. (2012). *PDDRA: A new pre-fetching based dynamic data replication algorithm in data grids*. *Future Generation Computer Systems*, 28(4), 666–681.
- Schintke, F., & Reinefeld, A. (2003). *Modeling replica availability in large data grids*. *Journal of Grid Computing*, 1(2), 219–227.
- Stockinger, H., Samar, A., Holtman, K., Allcock, B., Foster, I., & Tierney, B. (2002). *File and object replication in data grids*. *Cluster Computing*, 5(3), 305–314.
- Vazhkudai, S., Tuecke, S., & Foster, I. (2001). *Replica selection in the globus data grid*. *First IEEE/ACM International Symposium on Cluster Computing and the Grid*, CCGrid, 16-113.
- Yuan, Y., Wu, Y., Yang, G., & Yu, F. (2007). *Dynamic data replication based on local optimization principle in data grid*. *IEEE 6TH International Conference Grid and Cooperative Computing*, 815-822.
- Zhao, W., Xu, X., Wang, Z., Zhang, Y., & He, S. (2010). *A Dynamic Optimal Replication Strategy in Data Grid Environment*. *IEEE International Conference on Internet Technology and Applications (ITAP 2010)*, 1-4.