International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:8, No:8, 2014

Weighted Data Replication Strategy for Data Grid Considering Economic Approach

N. Mansouri, A. Asadi

***Abstract*—**Data Grid is a geographically distributed environment that deals with data intensive application in scientific and enterprise computing. Data replication is a common method used to achieve efficient and fault-tolerant data access in Grids. In this paper, a dynamic data replication strategy, called Enhanced Latest Access Largest Weight (ELALW) is proposed. This strategy is an enhanced version of Latest Access Largest Weight strategy. However, replication should be used wisely because the storage capacity of each Grid site is limited. Thus, it is important to design an effective strategy for the replication replacement task. ELALW replaces replicas based on the number of requests in future, the size of the replica, and the number of copies of the file. It also improves access latency by selecting the best replica when various sites hold replicas. The proposed replica selection selects the best replica location from among the many replicas based on response time that can be determined by considering the data transfer time, the storage access latency, the replica requests that waiting in the storage queue and the distance between nodes. Simulation results utilizing the OptorSim show our replication strategy achieve better performance overall than other strategies in terms of job execution time, effective network usage and storage resource usage.

***Keywords*—**Data grid, data replication, simulation, replica selection, replica placement.

I. Introduction

IN the increasing request of scientific and large scale business application, huge amount of data are produced and spread for using by users around the world [1]-[3]. It is difficult and inefficient to store such large amounts of data using a centralized storage. Grid technology is the best solution to this kind of problem. The main objective of the Grid Project is to provide sharing of computing and storage resources by users located in different part of the world. Kesselman and Foster in 1998 defined a Grid as follows ''A Computational Grid is a hardware and software infrastructure that provides dependable, consistent, pervasive, and inexpensive access to high-end computational capabilities'' [4]. Later in 2002 they improved the previous definition in [5] as follows ''A system that coordinates resources that are not subject to centralized control, using standard, open, general purpose protocols and interfaces to deliver non-trivial qualities of services''. Grid can be divided as two parts, Computational Grid and Data Grid. Computational Grids are used for computationally intensive applications that require small amounts of data. But, Data Grids deals with the applications that require studying and analyzing massive data sets [6]-[9]. Replication technique is one of the major factors affecting the performance of Data Grids by replicating data in geographically distributed data stores. The motivation for replication is how to enhance data availability, accessibility, reliability, and scalability. There are three key issues in all the data replication algorithms as follows [10]:

- Replica selection: Process of selecting replica among other copies that are spread across the Grid.
- Replica placement: Process of selecting a Grid site to place the replica.
- Replica management: Process of creating or deleting replicas in Data Grid.

Meanwhile, even though the memory and storage size of new computers are ever increasing, they are still not keeping up with the request of storing large number of data. The major challenge is a decision problem i.e. how many replicas should be created and where replicas should be stored. Hence methods needed to create replicas that increase availability without using unnecessary storage and bandwidth. In this work a novel dynamic data replication strategy, called Enhanced Latest Access Largest Weight (ELALW) is proposed. The ELALW strategy improves proposed algorithm (LALW) in [11]. According to the previous works, although LALW makes some improvements in some metrics of performance like mean job time, it shows three deficiencies:

(1) The LALW determines in which region the replica has to be placed and how many replica has to be placed. But LALW doesn't determine in which site within the region the file has to be placed. ELALW places replicas in two stages. In the first stage ELALW like LALW determines how many replicas have to be placed in each region. The second stage is to place the replica in the Best Storage Element (BSE) within the region. To select the BSE, ELALW finds SE with minimum Value-SE (VSE) in the region. In the calculation of VSE the frequency of requests of the replica and the last time the replica was requested are considered. These parameters are important because they give an indication of the probability of requesting the replica again.

(2) In replica replacement step using LFU strategy may delete some valuable files that may not be available in local region and may be needed in future. Therefore, such deletions will result in a high cost of transfer. ELALW considers three important factors into replacement decision: the number of requests in future based on

N. Mansouri is with the Department of Computer Science, Shahid Bahonar University of Kerman, 22 Bahman bolvar, Kerman, Iran (corresponding author to provide phone: +98-915-3624299; fax: 03412111865; e-mail: najme.mansouri@gmail.com).

A. Asadi was with the Department of Computer Science, Shahid Bahonar University of Kerman, Kerman, Iran (e-mail: amir.asadi@gmail.com).

Economic Model, the size of replica, the number of copies of the file.

(3) In replica selection step, LALW considers transfer time, this factor is not sufficient. The response time is a key parameter that influences the replica selection and thus the job turnaround time. ELALW strategy selects the best replica location for the users' running jobs by considering new parameters besides the data transfer time, namely, the storage access latency, waiting time in the storage queue and distance between nodes. Typically, the operating system dispatches the I/O requests in order to improve system performance. Scheduling can be implemented by keeping a queue of requests for the storage device. Thus, the storage media speed and the number of requests in queue has an influence on the average response time experienced by applications. So, the storage access latency is the delayed time for the storage media to perform the requests and this delayed time depends on the file size and storage type.

The proposed algorithm is implemented by using a data Grid simulator, OptorSim developed by European Data Grid project. The simulation results show that our proposed algorithm has better performance in comparison with other algorithms in terms of job execution time, effective network usage and storage resource usage.

The rest of the paper is organized as follows: Section II gives an overview of pervious work on data replication in Data Grid. Section III presents the novel dynamic data replication strategy. We show and analyze the simulation results in Section IV. Finally, Section V concludes the paper and suggests some directions for future work.

II. Related Works

Foster and Ranganathan [12], proposed six distinct replica strategies: No Replica, Best Client, Cascading Replication, Plain Caching, Caching plus Cascading Replica and Fast Spread) for multi-tier Data Grid. They also introduced three types of localities, namely:

- Temporal Locality: The files accessed recently are much possible to be requested again shortly.
- Geographical Locality: The files accessed recently by a client are probably to be requested by adjacent clients, too.
- Spatial Locality: The related files to recently accessed file are likely to be requested in the near future.

These strategies evaluated with different data patterns: first, access pattern with no locality. Second, data access with a small degree of temporal locality and finally data access with a small degree of temporal and geographical locality. The results of simulations indicate that different access pattern needs different replica strategies. Cascading and Fast Spread performed the best in the simulations. Also, the authors combined different scheduling and replication strategies.

Tang et al. [13] presented Simple Bottom-Up (SBU) and Aggregate Bottom-Up (ABU) strategies to improve the average data access response time for a multi-tier data grid. The main idea of the two strategies is to store a replica to nodes close to its requesting clients when the file's access rate is higher than a pre-defined threshold. SBU uses the file access history for each node, but ABU aggregates the file access history for a system. With ABU, a node transmits aggregated historical access records to its top tiers, and the top tiers do the same until these records reach the root. The results show that ABU improves job response time and bandwidth consumption better than those of SBU because its aggregation capability.

Shorfuzzaman et al. [14] proposed a new dynamic replica placement algorithm, Popularity Based Replica Placement (PBRP), for hierarchical data grids which is guided by file "popularity". The effectiveness of PBRP algorithm depends on the careful selection of a threshold value that relates to the popularity of files. They also presented an adaptive version of this strategy that find the threshold dynamically using such factors as data request arrival rates and available storage capacities at the replica servers. PBRP improves on ABU algorithm by making replicas accessible nearer to clients with lower access counts (which don't exceed the threshold value). The results of simulations show that their proposed algorithms can shorten job execution time significantly and reduce bandwidth consumption compared to other dynamic replication algorithms.

Andronikou et al. [15] proposed a set of interoperable new data replication strategies that take into account the infrastructural constraints as well as the 'importance' of the data. The presented system is scalable and the strategies can be easily implemented on a Grid environment to provide fast execution. The proposed QoS-aware dynamic replication strategy determines the number of replicas required based on data request, content importance and requested QoS. It also places of the new replicas within the Grid environment according to the network bandwidth and the overhead that the replication technique presents. It can handle the dynamicity of the Grid system by increasing or decreasing the set of data replicas based on the number and the geography of the data demands.

Lee et al. [16] presented an adaptive data replication strategy for a star-topology Data Grid, called the Popular File Replicate First algorithm (PFRF). It periodically computes file access popularity to track the changes of users' access behaviors, and then replicates popular files to suitable clusters/sites to adapt to the variation. They considered several types of file access behaviors, including Zipf-like, geometric, and uniform distributions, to evaluate PFRF. The simulation results demonstrate that PFRF can reduce average job turnaround time and bandwidth consumption.

Saadat et al. [17] presented a new dynamic data replication strategy which is called Pre-fetching based Dynamic Data Replication Algorithm in Data Grids (PDDRA). PDDRA predicts future requires of Grid sites and pre-replicates them before needs are requested. This prediction is done based on the past file access history of the Grid sites. So when a Grid site requests a set of files, it will get them locally. The simulation results show that this strategy improves in terms of

job execution time, effective network usage, number of replications, hit ratio and percentage of storage filled.

Taheri et al. [18] proposed a new Bee Colony based optimization strategy, called Job Data Scheduling using Bee Colony (JDS-BC). JDS-BC has two collaborating operations to efficiently schedule jobs onto computational elements and replicate data sets on storage elements in a system so that the two independent, and in many cases conflicting, objectives (i.e., makespan and transfer time of all data files) of such heterogeneous systems are concurrently decreased. Three tailor-made test Grids varying from small to large are applied to evaluate the performance of JDS-BC and compare it with other strategies. Results showed that JDS-BC's superiority under different operating scenarios. JDS-BC also presented a balanced decision making behavior, where it occasionally relaxes one of its objectives (e.g., transfer time) to obtain more from optimizing the other one (e.g., makespan).

Mansouri and Dastghaibyfard [19] presented a Dynamic Hierarchical Replication (DHR) strategy that store replica in suitable sites where the particular file has been accessed most, instead of storing file in many sites. It also decreases access latency by selecting the best replica when different sites hold replicas. The proposed replica selection strategy chooses the best replica location for the users' running jobs by considering the replica requests that waiting in the storage and data transfer time. The simulation results show, it has less job execution time in comparison with other strategies especially when the Grid sites have comparatively small storage size.

Park et al. [20] presented a Bandwidth Hierarchy based Replication (BHR) which decreases the data access time by maximizing network-level locality and avoiding network congestions. They divided the sites into several regions, where network bandwidth between the regions is lower than the bandwidth within the regions. So if the required file is placed in the same region, its fetching time will be less. BHR strategy has two deficiencies, first it terminates, if replica exists within the region and second replicated files are placed in all the requested sites not the appropriate sites. BHR strategy has good performance only when the capacity of storage element is small. Modified BHR [21] is an extension of BHR [20] strategy which replicates a file that has been accessed most and it may also be used in near future.

A replication algorithm for a 3-level hierarchy structure and a scheduling algorithm are proposed in [22]. They considered a hierarchical network structure that has three levels. In their replica selection method among the candidate replicas they selected the one that has the highest bandwidth to the requested file. Similarly, it uses the same technique for file deletion. This leads to a better performance comparing with LRU (Least Recently Used) method. For efficient scheduling, their algorithm selects the best region, LAN and site respectively. Best region (LAN, site) is a region (LAN, site) with most of the requested files.

III. Proposed Replication Algorithm

In this section, first network structure is described and then the LALW and ELALW algorithms are presented.

A. Network Structure

Fig. 1 shows the hierarchical architecture which has two levels similar to what is given in [11]. We locate group of the Grid sites on the same network region. A network region is a network topological space where sites are placed closely. There is a region master responsible for replica management. There is a region header used to manage the site information in a region. The region master gets the information of accessed files from all headers. Each site has the history of files accessed in that site. The access history provides the details of FileId, RegionId and the Time, which shows that the file has been accessed by a site placed in the region (RegionId) at a particular time. Each site sends its access history to its region header regularly. All information in the same region will be aggregated and summarized by the region header. The region master provides the global information based on the file and the number of time it has been accessed.

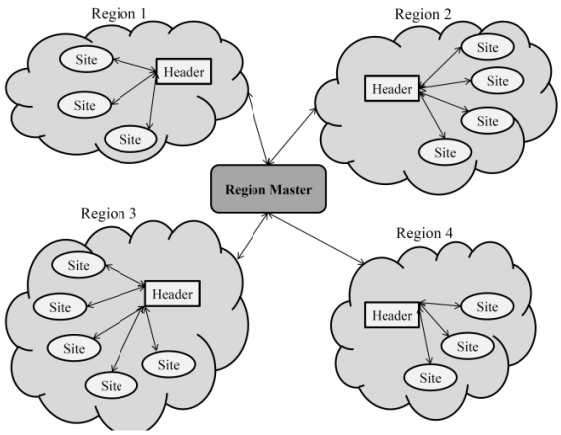

Fig. 1 The Hierarchical architecture

B. LALW Strategy

In [11] LALW strategy is proposed where the replicas are created dynamically based on the weights and placed in the regions. LALW strategy is briefly explained and more information about LALW algorithm can be found in [11]. Replica creation in LALW strategy has two steps. First step determines which file to be replicated and the second step identifies the number of replicas. In first step region master collects the information from the region header at a constant time interval. Information collected at different time interval has different weights. The rule of setting weight employs the concept of Half-life which is mentioned in many fields, such as physics, chemistry. Here the weight indicates the quantity and the time interval. It assigns larger weight to the recently accessed files and less weight to the older files. Access Frequency (AF) represents the importance for access history in various time intervals. The Access Frequency for file F is defined by the equation:

$$AF(f) = \sum_{t=1}^{N_T} (a_f^t \times 2^{-(N_T - t)}), \forall f \in F \quad (1)$$

where N_T is the number of time interval passed, F is the set of files that have been requested and a_i^j represents the number of accesses for the file i at time interval j. The file with maximum AF is a popular file. The second step is to find the number of replicas needed. The number of replicas can be calculated by comparing the average access frequency of the popular file with all other files. Assuming p is the popular file. Then the average access frequency of the popular file p is given by

$$AF_{avg}(p) = \frac{AF(p)}{N_T} \tag{2}$$

The average Access Frequency of all other files is given by

$$AF_{avg}(f) = \frac{[AF(f)]_{sum}}{N_F \times N_T}, \forall f \in F \tag{3}$$

$AF(f)_{sum}$ indicates the sum of AF for all files requested and $N_F=|F|$ is the number of different files that have been requested. The number of replicas needed for the popular file is calculated as

$$Num_{system}(p) = \left\lceil \frac{AF_{avg}(p)}{AF_{avg}(f)} \right\rceil \tag{4}$$

The region master requests to every region header to obtain the information about the popular file. The information involves the region details and the number field. These files still have various weights based on various time intervals. The Region master determines $AF(p)$ in different regions after collecting the information for popular file from region headers. Then it sorts regions in descending order according to the $AF(p)$. The first region in the sorted list has the highest priority to have the replicas. Number of replicas to be placed at region c is

$$Num_c(p) = \left\lceil n \times \frac{AF_c(p)}{[AF_c(p)]_{sum}} \right\rceil, c = 1,2,...N \tag{5}$$

where n is the number of replicas needed to be replicated. $AF_c(p)$ represents the AF for the popular file p in region c, $[AF_c(p)]_{sum}$ is the sum of $AF_c(p)$ for all regions.

Finally, if there is no space to store the replica Least Frequently Used (LFU) file is deleted from the site.

C. ELALW Strategy

We describe ELALW strategy also in four sections.

Replica Creation: At intervals, the proposed algorithm like LALW collects the information about accessed files from all headers. By calculating the product of weight and the number of accesses for a file, it considers a more precise metric to determine a popular file for replication. Then it calculates the number of replicas needed from (4).

Replica Placement: Replica placement has two stages. In the first stage ELALW like LALW determines how many replicas have to be placed in each region by using (5). The second stage is to place the replica in the Best Storage Element (BSE) within the region. To select the BSE, ELALW finds SE with minimum Value-SE (VSE) in the region. In the calculation of VSE the frequency of requests of the replica and the last time the replica was requested are considered. These parameters are important because they give an indication of the probability of requesting the replica again.

$$VSE = (CT - LT) + \frac{1}{FR} \tag{6}$$

where CT is the current time, LT the last request time of the replica, and FR the frequency of requests of the replica.

Replica Selection: When different sites have replicas of file, there is a significant benefit realized by selecting the best replica. Replica selection decides which replica location is the best for the Grid users. Four factors are used to choose a best replica:

- Storage Access Latency

The storage media speed and size of requests queue have a key role in the average response time. T_1 can be calculated by the following equation:

$$T_1 = \frac{FileSize(MB)}{StorageSpeed(MB/Sec)} \tag{7}$$

- Transfer Time

Transfer time is defined as the data transmission via a wide area network, which depends on the network bandwidth and the size of the file. It is computed by the following equation:

$$T_2 = \frac{FileSize(MB)}{Bandwidth(MB/Sec)} \tag{8}$$

- Waiting Time in the Storage Queue

Each storage media has some requests at the same time and the storage can perform only one request at a time. So, one has to wait for all the previous requests in the storage queue. T_3 can be defined by the following equation:

$$T_3 = \sum_{i=0}^{n} T_1 \tag{9}$$

where n is number of requests waiting in the queue.

- Distance between nodes

$D(x,y)$ represents network distance between nodes x and y. Computed using the number of hops with a trace route command. To reduce the cost, distance information can be stored when a replica is checked for the first time.

$$T = T_1 + T_2 + T_3 \tag{10}$$

$$F(x, y) = w_1 \times T + w_2 \times D(x, y) \quad (11)$$

This function can be tailored, because it is defined as a weighted combination of the two former metrics. The proper weights (w_1, w_2) have been obtained empirically.

Replica Replacement: If enough space for replication does not exist, a list of replicas needs to be removed from the storage. But what if that list of replicas that are to be deleted are more valuable than the new replica? ELALW strategy stores only the important replicas while the other less necessary replicas are replaced with more important replicas. ELALW strategy removes that list only if the value of that list is smaller than the value of the requested replica. It considers three factors in replacement step: the number of requests in future, the size of the replica, and the number of copies of the file. The number of requests represents how many times the replica has been requested by its node. Since the storage space is the main problem in the Data Grid, the size of replica is also a key parameter in deciding if the replica should be stored. The value of list (VL) is given by the equation

$$VL = \frac{\sum_{i=1}^{n} NR_i}{\sum_{i=1}^{n} S_i} + \frac{1}{NC_i} \quad (12)$$

NR_i is predicted number of times replica *i* will be requested based on Economic Model, S_i is the size of replica *i* in the list, and NC_i is the number of copies of the replica *i*.

Economic Model uses an evaluation function that could estimate a file's future revenue based on its past access frequency. File access history can be represented as a random walk in the space of file identifiers. In the random walk, the identifier of the next requested file is sum of the current identifier and a step, the value of which is given by a binomial distribution. The evaluation function *E (f, r, n)* is [23]:

$$E(f, r, n) = \sum_{i=1}^{n} p_i(f) \quad (13)$$

$$n = r\frac{\delta t'}{\delta t} \quad (14)$$

where t' is the time interval when the past *r* requests were made and *t* is the future time interval for which we intend to do prediction. Then the predicted number of times a file will be requested in the next *n* requests based on the past *r* requests in the history are given from (13).

The value of replica (*VR*) is given by the equation

$$VR = \frac{NR}{S} + \frac{1}{NC} \quad (15)$$

1. We know that storing multiple the file copies in the same SE does not enhance the file availability, because if the SE fails, all the files on the SE will fail in the same time. This can damage the system level data availability because such unnecessary replications will waste the storage space in the SE. Therefore, if the new replica already exists in the BSE, it will not be replicated again.
2. If enough storage space exists in the BSE, the new replica is stored.
3. When there is not enough storage, the potential candidates for replacement will be chosen according to their value which is to give (12). Generate a list of replicas that are available in BSE in increasing order based on their VR. If two or more replicas have the same VR, they are sorted randomly. Then it fetches the replicas from the sorted replica list in order and add it into candidate list until the accumulative file size of candidate list are greater than or equal to the new replica size.
4. VL is calculated for candidate list which is given from previous step.
5. VR is calculated for the new replica.
6. If VL < VR then store new replica by deleting the candidate list.

Fig. 2 explains the replacement strategy.

IV. Experiments

In this section, simulation tool, configuration, experiment results and discussion are described respectively.

A. Simulation Tool

OptorSim was developed to simulate the structure of a real Data Grid for evaluating various replication strategies. OptorSim is the project of EDG, a Java-based simulation language. Fig. 3 shows the architecture of the Data Grids simulated by the OptorSim simulator [24], [25]. OptorSim assumes that a Grid consists of several sites, each of which contains zero or more Computing Elements (CEs) which execute jobs and zero or more Storage Elements (SEs) which store files. Resource Broker (RB) accepts job submission from users and dispatches each job to proper site according to the scheduling algorithm, which collect some information to make an optimal decision. Replica Manager (RM) at each site controls data transferring and provides a mechanism for accessing the Replica Catalog. Replica Optimizer (RO) within RM implements the replication algorithm shown in Fig. 3 (b).

B. Configuration

With OptorSim, it is possible to simulate any Grid topology and replication strategy. So OptorSim code has been modified to implement the hierarchical structure, since it uses a flat network structure. The Grid topology of simulated platform is given in Fig. 4. It is assumed the network has four regions and each one has three sites. Node 8 has the most capacity to store all the master files at the beginning of the simulation. The storage capacity of all other sites is 40GB. The connection bandwidth is 100 Mbps. We ran the simulation with 1600 jobs. The number of file accessed per job on average is 16 and job delay is 2500 ms. Each data file to be accessed is 2 Gbyte. To simplify the requirements, we assumed that the data is read-only.

1. If new replica exists in the BSE
   Do nothing
2. New replica does not exist in the BSE
   BSE has enough free space
      Store new replica in BSE
3. New replica does not exist in the BSE
   BSE does not have enough free space
      Sort the files in BSE by the file value VR (equation (15)) in ascending order
      Fetch the files the sorted file list in order and add it into candidate list until the accumulative file size of the candidate files are greater or equal to the new replica
4. VL is calculated for candidate list
5. VR is calculated for the new replica
6. If VL < VR then store the new replica by deleting the candidate list

Fig. 2 Replacement strategy

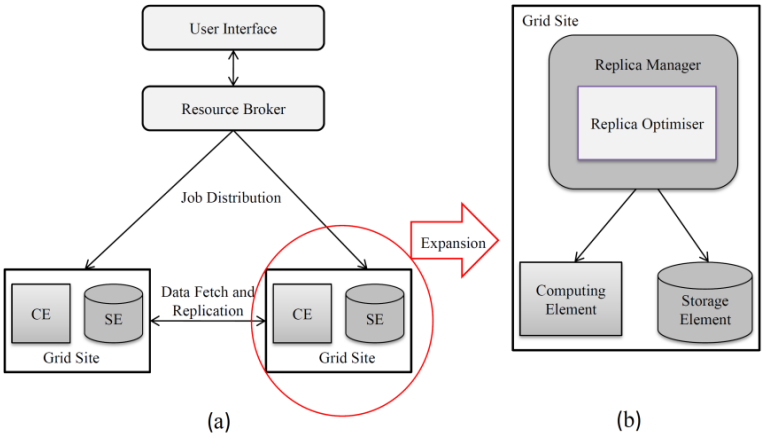

Fig. 3 (a) OptorSim architecture (b) An expanded illustration of grid site

C. Simulation Results and Discussion

The three performance evaluation metrics used for the simulation are shown below.

- Mean Job Execution Time is determined as the total time to run all the jobs, divided by the number of jobs completed.
- Storage Resource Usage shows the percentage of available spaces that are used.
- Effective Network Usage is the ratio of files transmitted to files requested.

We evaluate and compare the performance of ELALW algorithm with eight replication algorithms; No Replication (NR), Least Frequency Used (LFU), 3-Level Hierarchical Algorithm (3LHA), Bandwidth Hierarchy based Replication algorithm (BHR), Modified BHR (MBHR), Popularity Based Replica Placement (PBRP), Dynamic Hierarchical Replication (DHR), and Latest Access Largest Weight (LALW) algorithm. In No Replication strategy files are accessed remotely. When storage is full, LFU deletes least frequency accessed files. The 3LHA considers a hierarchical network structure that has three levels. Bandwidth is an important factor for replica selection and deletion. The BHR algorithm stores the replicas in a site that has a high bandwidth and replicates those files that are likely to be requested soon within the region. The MBHR algorithm replicates the files within the region in a site where file has the highest access. The PBRP is an adaptive technique to control the degree of replication as the file request rate fluctuates.

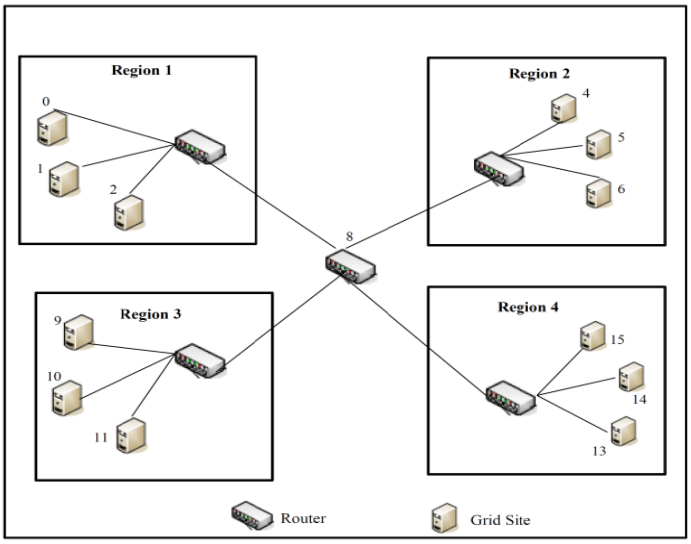

Fig. 4 Grid topology in the simulation

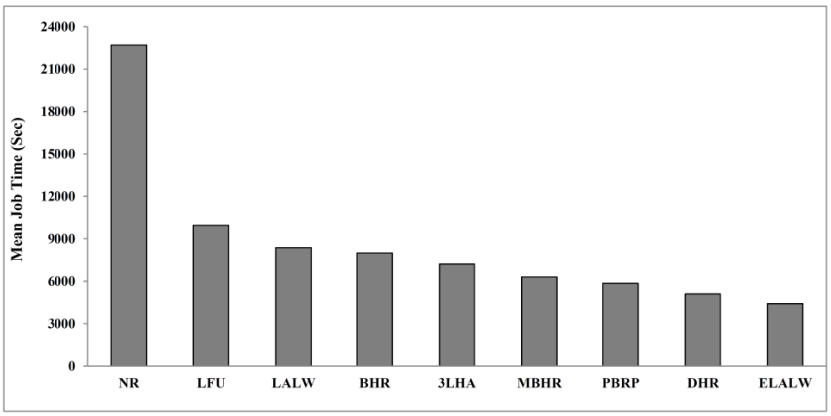

Fig. 5 Mean job execution time for various replication algorithms

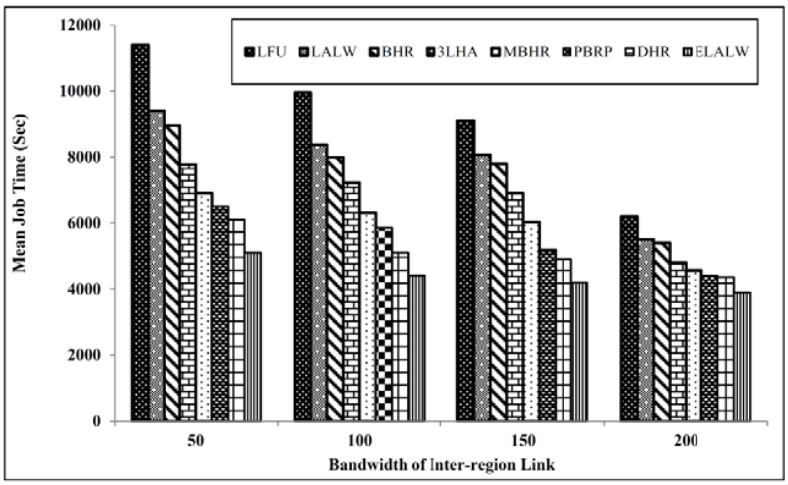

Fig. 6 Mean job time with varying bandwidth

The DHR algorithm places replicas in appropriate sites i.e. best site that has the highest number of access for that particular replica. It also minimizes access latency by selecting the best replica by considering the replica requests that waiting in the storage and data transfer time.

In Fig. 5, the execution time of ELALW is smaller than other strategies. ELALW improves the mean job execution time by selecting the best replica location for execution jobs with considering number of requests that waiting in the storage queue, data transfer time, storage access latency and the distance between nodes. Obviously, the No Replication strategy has the worst performance as all the files requested by jobs have to be transferred from CERN. BHR algorithm improves data access time by avoiding network congestions. If the available storage for replication is not enough, BHR applies 2-step decision process. First one is avoiding duplication. Secondly, BHR takes account of popularity of files. 3LHA performs better than BHR because it considers the differences between intra-LAN and inter-LAN communication. PBRP adjusts the threshold value based on the varying arrival rate which leads to the creation of more replicas compared to MBHR which decreases the execution time.

Fig. 6 shows the mean job time of replication algorithms for varying inter region bandwidth. When we set narrow bandwidth on the inter-region link, our strategy outperforms other strategy considerably. According to the temporal and geographical locality, ELALW places the replica in the best site i.e. best site that has minimum Value-SE (VSE) for that particular replica. Therefore it can reduce the intercommunications between different region Grids.

Fig. 7 displays the mean job time based on changing number of jobs for eight algorithms. It is clear that as the job number increases, ELALW is able to process the jobs in the lowest mean time in comparison with other methods. Since it will not delete those file that have a high transferring time. It uses the Economic Model to decide and delete those files that are not beneficial in the future and replaces them with files that are more beneficial in the future. It is similar to a real Grid environment where a lot of jobs should be executed.

Data replication takes time and consumes network bandwidth.

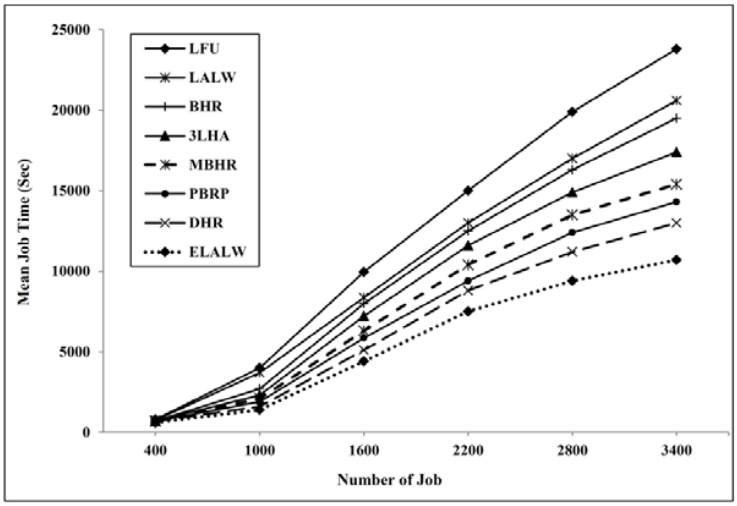

Fig. 7 Mean job time based on varying number of jobs

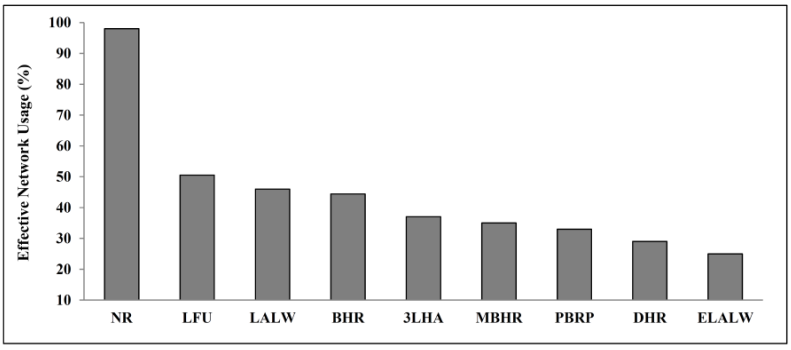

Fig. 8 Effective network usage with sequential access pattern generator

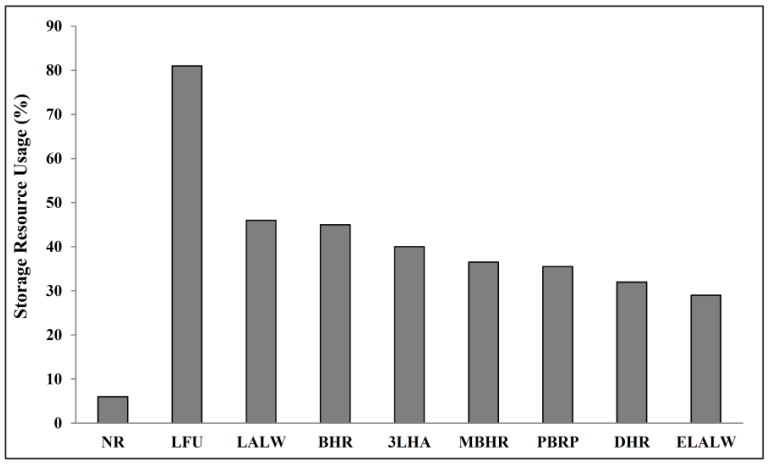

Fig. 9 Storage resources usage

However, performing no replication has been demonstrated to be ineffective compared to even the simplest replication strategy. So, a good balance must be discovered, where any replication is in the interest of reducing future network traffic. ENU is used to estimate the efficiency the network resource usage. Effective Network Usage (E_{enu}) is given from [26]:

$$E_{enu} = \frac{N_{rfa} + N_{fa}}{N_{lfa}} \quad (16)$$

where N_{rfa} is the number of access times that CE reads a file from a remote site, N_{fa} is the total number of file replication operation, and N_{lfa} is the number of times that CE reads a file locally. The effective network usage ranges from 0 to 1. A lower value represents that the network bandwidth is used more efficiently. Fig. 8 shows the comparison of the Effective Network Usage of the five replication strategies for the sequential access pattern. The ENU of ELALW is lower about 48% compared to the LALW strategy. The main reason is that Grid sites will have their needed files present at the time of need, hence the total number of replications will decrease and total number of local accesses increase. The ELALW is optimized to minimize the bandwidth consumption and thus decrease the network traffic. A low ENU value in the case of ELALW indicates that it is good at putting files in the right places. The No Replication strategy operates the worst and consumes the maximum network bandwidth available in the network.

Fig. 9 depicts the storage resource usage. The storage resource usage of the No replication is best because in this case the data is stored only in one location where the files are produced initially. But ELALW algorithm creates replicas dynamically in advance. Instead of storing files in many sites, they can be stored in a particular site so that the storage usage can be reduced.

V. Conclusion

Data Grid is a very important and useful technology to process the large number of data produced by data-intensive computing applications. A common solution to improve availability and file access time in a Data Grid is to replicate the data. In this paper we propose a novel dynamic data replication method which is based on LALW. According to the previous works, although LALW makes some improvements in some metrics of performance like storage usage, it shows some deficiencies. First LALW determines in which region header the replica has to be placed and how many replica has to be placed. But it doesn't determine in which site within the region the file has to be placed. Second, if the available storage for replication is not enough Least Frequently Used (LFU) file is deleted from the site. In this step using LFU may delete some valuable files that may be needed in future. Finally, the response time is estimated by considering the data transfer time only in selecting the required replicas. The Enhanced Latest Access Largest Weight (ELALW) algorithm is designed to lessen these weaknesses. ELALW stores the replicas in the best site where the file has been accessed for the most time instead of storing files in many sites. It also improves access latency by selecting the best replica when various sites hold replicas. The proposed replica selection selects the best replica location from among the many replicas based on response time that can be determined by considering the data transfer time, the storage access latency, the replica requests that waiting in the storage queue and the distance between nodes. Restricted by the storage capacity, it is essential to present an effective strategy for the replication replacement task. ELALW replaces replicas based on the number of requests in future, the size of the replica, and the number of copies of the file. To evaluate the efficiency of our data replication algorithm, we tested it with Data Grid simulator, OptorSim. Mean Job Time, Effective Network Usage and Storage Usage were used as the performance evaluation metrics. From the simulation experiment, it can be concluded that ELALW algorithm can achieve a significant improvement of performance over former similar work, because ELALW replicates at regular intervals and stores them in appropriate sites. Therefore it reduces

unnecessary replication. In the future, we would like to consider the set of QoS factors taken into account for dynamic replication, including both service provider and client-related requirements. We also try to investigate dynamic replica maintenance issues such as replica consistency. Finally, we plan to test our simulation results on real Data Grid.

ACKNOWLEDGMENTS

This work was supported by Science Technology Park (STP).

REFERENCES

[1] A. Folling, C. Grimme, J. Lepping and A. Papaspyrou, "Robust load delegation in service Grid environments," *IEEE Transactions on Parallel and Distributed Systems*, vol. 21, pp. 1304-1316, 2010.
[2] O. Sonmez, H. Mohamed, and D. Epema, "On the benefit of processor coallocation in multicluster Grid systems," *IEEE Transactions on Parallel and Distributed Systems*, vol. 21, pp. 778-789, 2010.
[3] Li, H., "Realistic workload modeling and its performance impacts in large-scale Escience Grids," *IEEE Transactions on Parallel and Distributed Systems*, vol. 21, pp. 480-493, 2010.
[4] I. Foster and C. Kesselman, "The Grid: blueprint for a new computing infrastructure," *Morgan Kaufmann*, 2004.
[5] I. Foster, C. Kesselman and S. Tuecke, "The anatomy of the Grid," 2001
[6] GriPhyN: The Grid physics network project, 12 July 2010. http://www.griphyn.org.
[7] R.S. Chang and M.S. Hu, "A resource discovery tree using bitmap for Grids," *Future Generation Computer Systems*, vol. 26, pp. 29-37, 2010.
[8] J. Wu, X. Xu, P. Zhang and C. Liu "A novel multi-agent reinforcement learning approach for job scheduling in Grid Computing," *Future Generation Computer Systems*, vol. 27, pp. 430-439, 2011.
[9] S. Ebadi, and L.M. Khanli, "A new distributed and hierarchical mechanism for service discovery in a Grid environment," *Future Generation Computer Systems*, vol. 27, pp. 836-842, 2011.
[10] W. Shih, C.T. Yang and S.S. Tseng, "Using a performance-based SKELETON to implement divisible load applications on Grid Computing environments," *Journal of Information Science and Engineering*, vol. 25, pp. 59-81, 2009.
[11] R.S. Chang and H.P. Chang, "A dynamic data replication strategy using access weights in Data Grids," *Journal of Supercomputing*, vol. 45 (3), pp. 277-295, 2008.
[12] K. Ranganathan and I. Foster, "Identifying dynamic replication strategies for a high performance Data Grid," *in Proceedings of the Second International Workshop on Grid Computing*, pp. 75-86, 2001
[13] M. Tang, B.S. Lee, C.K. Yao and X.Y Tang, "Dynamic replication algorithm for the multi-tier Data Grid," *Future Generation Computer Systems*, vol. 21 (5), pp. 775-790, 2005.
[14] M. Shorfuzzaman, P. Graham, R. Eskicioglu, "Adaptive popularity-driven replica placement in hierarchical data grids," *The Journal of Supercomputing*, vol. 51, pp. 374–392, 2010.
[15] V. Andronikou, K. Mamouras, K. Tserpes, D. Kyriazis and T. Varvarigou, "Dynamic QoS-aware data replication in Grid environments based on data "importance"," *Future Generation Computer Systems*, vol. 28 (3), pp. 544-553, 2012.
[16] M.C. Lee, F.Y. Leu, and Y. Chen, "PFRF: An adaptive data replication algorithm based on startopology Data Grids," *Future Generation Computer Systems*, 2011.
[17] N. Saadat, A.M. Rahmani, "PDDRA: A new pre-fetching based dynamic data replication algorithm in Data Grids," *Future Generation Computer Systems*, 2011.
[18] J. Taheri, Y.C. Lee, A.Y. Zomaya and H.J. Siegel, "A Bee Colony based optimization approach for simultaneous job scheduling and data replication in Grid environments," Computers & Operations Research, 2011.
[19] N. Mansouri, G.H. Dastghaibyfard, "A dynamic replica management strategy in Data Grid," *Journal of Network and Computer Applications*, 2012.
[20] S.-M.Park, J.-H.Kim, Y.-B.Go and W.-S. Yoon, "Dynamic Grid replication strategy based on internet hierarchy," *in International Workshop on Grid and Cooperative Computing*, vol. 1001, pp. 1324-1331, 2003.
[21] K. Sashi and A. Thanamani, "Dynamic replication in a Data Grid using a modified BHR region based algorithm," *Future Generation Computer Systems*, vol. 27 (2), pp. 202-210, 2011.
[22] A. Horri, R. Sepahvand, and G.H. Dastghaibyfard, "A hierarchical scheduling and replication strategy," *International Journal of Computer Science and Network Security*, vol.8, 2008.
[23] W.H. Bell, D.G. Cameron, L. Capozza, A. Paul Millar, K. Stockinger, F. Zini, "Simulation of Dynamic Grid Replication Strategies in OptorSim," *in Proc. of the ACM/IEEE Workshop on Grid Computing* (Grid 2002) 2002.
[24] D.G. Cameron, A.P. Millar, C. Nicholson, R. Carvajal-Schiaffino, F. Zini and K. Stockinger, "Optorsim: A simulation tool for scheduling and replica optimization in Data Grids," *in International Conference for Computing in High Energy and Nuclear Physics* (CHEP 2004), 2004.
[25] OptorSim–A Replica Optimizer Simulation: http://edg-wp2.web.cern.ch/edgwp2/ optimization/optorsim.html.
[26] W.H. Bell, D.G. Cameron, R. Carvajal-Schiaffino, A.P. Millar, K. Stockinger, and F. Zini, "Evaluating Scheduling and Replica Optimization Strategies in Data Grid," *IEEE*, 2003.