TELKOMNIKA Indonesian Journal of Electrical Engineering
Vol.12, No.4, April 2014, pp. 2860 ~ 2867
DOI: http://dx.doi.org/10.11591/telkomnika.v12i4.4737

Dynamic Data Grid Replication Algorithm Based on Weight and Cost of Replica

Zhongping Zhang[1,2]*, Chao Zhang[1], Mengfei Zuo[1], Zhiping Wang[3]
[1]College of Information Science and Engineering, Yanshan University, Qinhuangdao, Hebei 066004, China
[2]The Key Laboratory for Computer Virtual Technology and System Integration of Hebei Province, Qinhuangdao, Hebei 066004, China
[3]Foreign Language College of Dalian Jiaotong University, Dalian, Liaoning, 116052, China
*Corresponding author, e-mail: zpzhang@ysu.edu.cn

Abstract

Data Grid is composed of a large number of distributed computation and storage resources to facilitate the management of the huge distributed and sharing data resources efficiently. Dynamic replication can reduce the file storage time and use the grid resources effectively in a Data Grid environment. The Data Grid topology is divided into three layers: Regional level, LAN level, the grid site level. We have improved DHRA algorithm in three areas: replica selection, replica placement, replica replacement. Considering the weight and cost of replica, we propose dynamic Data Grid replication algorithm LWLC Based on Weight and Cost of Replica. At last, we use the OptorSim to prove the effectiveness of the LWLC.

Keywords: *data grid, hierarchical, dynamic replication, optorsim*

1. Introduction

At present, with the increase of demand of large-scale commercial applications and the social sciences, Large-scale data generated all the time in the world [1]. For example, high-energy physics, bioinformatics, earth observation, global climate change, image processing, data mining, data in these applications reached TB level or even PB level [2]. These data cannot be stored in some centralized sites and must be dispersed throughout the world. In this case, how to manage and store huge amount of data is very difficult and even impossible [3-4]. The Data Grid can effectively solve this problem; it tries to store these data in distributed sites, and provides retrieves for each application.

The network bandwidth is the most important factor that affects the performance of Data Grid, i.e. slow data access restrict the performance of data-intensive applications in Data Grid. the topological structure of Data Grid was proposed in 3LHA (3-Level Hierarchical Algorithm) algorithm [5], it has three levels: regions, LANs within each region, and sites within each LAN, the network bandwidth of the three levels are in ascending order. Regional level communicates through WAN, thus avoiding regional level access can effectively reduce access latency and improve the performance of Data Grid. We use data replication mechanism to address this problem. Data replication is another important optimization measure which used to manage the geographically distributed data. Effective data replication can bring lower bandwidth consumption, and can provide more available data. Data replication is divided into three major parts in this paper: (1) replica selection, i.e. the process of selecting the best replica from those replicas geographically spreaded across the Data Grid; (2) replica placement, i.e. the process of selecting the best SE (storage element) to store replica; (3) replica replacement, i.e. when the local SE does not have enough storage space to store replica, how to delete some files to store the new replica.

Based on DHRA (Dynamic Hierarchical Replication Algorithm) algorithm [6], We make three improvements: replica selection, replica placement, replica replacement. Then propose dynamicdata grid replication algorithm based on weight and cost of replica, called LWLC (Least weight and Least Cost). Simulation results with OptorSim show that LWLC gives better performance compared to other algorithms.

Received September 7, 2013; Revised November 7, 2013; Accepted November 21, 2013

2. Related Works

Currently, Data Grid researchers have a high interest in effective replication strategy. Here are several common scheduling algorithms.

Sang-Min Park, etc. proposed BHR (Bandwidth Hierarchy based Replication) algorithm [7] in 2004, the algorithm reduces data access latency mainly by increasing the local file access and avoiding network congestions. The Data Grid topology is divided into two levels: site level, region level. Network bandwidth between the regions is lower than the bandwidth between the sites. The algorithm replicates the replica to the site if there are enough space, or accesses the replica remotely if the replica available in the same region. Otherwise it tries to make available space by deleting files using LRU (Least Recently Used), then replicate the replica. This algorithm has two drawbacks: (1) it considers popularity of replicas at region level; (2) replicas are placed at all the requested sites, not only the appropriate site.

Ruay-shiung Chang, etc. proposed scheduling algorithm HCS (Hierarchical Cluster-based Scheduling) and data replication algorithm HRS (Hierarchical Replication Strategy) [8] in 2007, the two algorithm improved the efficiency of data storage in Data Grid. HCS uses a hierarchical scheduling and decreases searching time for the best computing site by using cluster information. HRS algorithm makes two improvements based on BHR algorithm, namely: (1) BHR checks all sites to select the best replica, but HRS algorithm selects the best within cluster, if the cluster does not exist then checks other clusters; (2) BHR uses the frequency of replicas in cluster level, while HRS algorithm uses the frequency of replicas in site level. HCS job scheduling algorithms along with HRS replication strategy has less job execution time in comparison to other scheduling algorithms and replication strategies.

A.Horri, etc. proposed 3LHA (3-Level Hierarchical Algorithm) algorithm in 2008. This paper relates to the job scheduling and dynamic data replication. They considered a hierarchical network structure that has three levels: region, LANs within each region, and sites within each LAN, the network bandwidth of the three levels is in ascending order. For efficient scheduling of any job, the algorithm determines most appropriate region (LAN, site) that holds most of the requested replicas(from size point of view), i.e. most of the requested replicas are available in that region(LAN, site). This can significantly reduce total transfer time and network traffic. The main weakness of 3LHA algorithm is it places replicas at all requested sites but not just the appropriate sites.

Najme Mansouri, etc. presented DHR (Dynamic Hierarchical Replication) algorithm [9] based on the 3LHA algorithm in 2012; it also used three-tier grid architecture. At replica selection, DHR selects the site that has least number of request; at replica placement, DHR creates a sorted list (by number of replica access) of all SE's that requested the particular file in region, then the replica will be placed in the first SE of the above sorted list.

Najme Mansouri, etc. proposed MLALW (Modify Latest Access Largest weight Strategy) [10] algorithm based on LALW (Latest Access Largest weight Strategy) [11] algorithm in 2012. MLALW deletes files by considering three important factors: least frequently used replicas, least recently used replicas and the size of the replica. MLALW stores each replica in an appropriate site, i.e. appropriate site in the region that has the highest number of access in future for that particular replica.

Najme Mansouri, etc. proposed a scheduling algorithm CSS (Combined Scheduling Strategy) and dynamic Data Grid replication algorithm DHRA algorithm in 2013. CSS first determines the region that has the most of the requested files (from the size point of view), i.e. most of the requested files are available in that region. Then, combined cost for each site within the best region is computed and the job will be assigned to the site with the Minimum Combined Cost. DHRA has three parts: (1) Replica Selection, it generates a list of replica that are available in other regions, and from this list it selects the replica that has the minimum Response Time which is calculated as the time elapsed for transferring a data file from one site to another. (2) Replica placement, the replica is placed at the SE that has most of requests in the region. (3) Replica Management, first deletes those files with minimum time for transferring, i.e. files that are both available in the current site as well as in the local LAN, second if space still insufficient, then repeats first step for each LAN in the current region, third if space still insufficient, then deletes the remaining files by using LRU method till space is available for replica.

These algorithms only consider certain aspects that affect the performance of Data Grid, but not consider comprehensively. Such as: replica replacement, some algorithms use LRU method to delete files and others use LFU method; replica selection, some algorithms only

consider the waiting time of copying a replica; replica placement, some algorithms only take into account the delay of network transmission.

3. Proposed Replication Algorithm

In this section, first the topology of Data Grid is described and then LWLC algorithm is presented.

3.1. Topology of Data Grid

Figure 1 shows the hierarchical architecture which has three levels similar to what is given in 3LHA algorithm. First level is region, they are connected via internet which has low bandwidth; second level contains local LANs within the region, they have a moderately higher bandwidth comparing to internet between them. Third level comprises the sites within each LAN, which have the highest bandwidth.

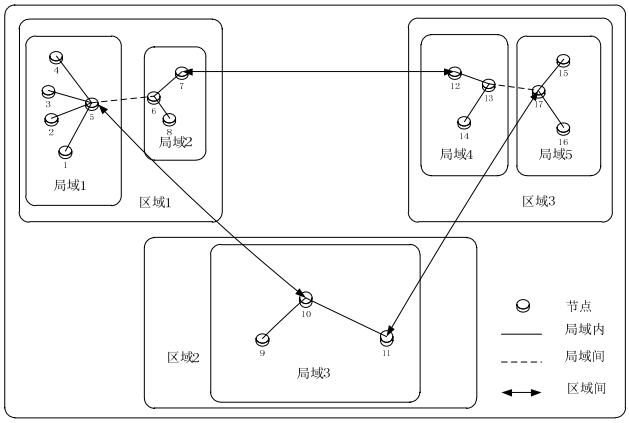

Figure 1. Data Grid Topology

3.2. LWLC Algorithm

When a job is dispatched to a site, it need copy the files that the job required but the SE don't have, this process is data replication. This paper presents dynamic Data Grid replication algorithm based on weight and cost replica, called LWLC. LWLC algorithm is proposed based on the DHRA, it improves on three aspects: replica selection, considering not only the waiting time of copying the replicas but also the transmission delay; replica placement, When there are several SEs in the region have the same number of requesting the particular replica, DHRA algorithm just randomly selects a SE, while LWLC algorithm selects the SE, it has the maximum number of different jobs that requested the particular replica, Thereby reducing the waiting time for multiple jobs; replica replacement: LWLC algorithm considers the weight of replica in different time interval and the cost of replicas.

3.2.1. Replica Selection

Select the optimal source SE that can reduce the transfer time. Affecting the transfer speed of replica mainly related to three aspects: the network bandwidth between source SE and destination SE, Physical copy speed of the source SE and the waiting time in the request queue. They can be measured with data transmission delay, storage access latency and waiting time in the request queue, respectively. Then we propose the T_t (Transfer Time), it can be defined by the following equation:

$$T_t = \frac{S_F(f)}{W_{rt}} + \frac{S_F(f)}{S_S} + \sum_{i=1}^{n} \frac{S_F(i)}{S_S} \quad (1)$$

Where $S_F(f)$ is the size of desired replica f, unit is MB; W_{rt} is the effective bandwidth between source SE and destination SE, unit is MB/S; $S_F(i)$ is the size of replica that wait in the requested queue of source SE, unit is MB/S; n is the number of replica which wait in the requested queue before desired replica f; S_s is the physical copy speed of source SE, unit is MB/S.

When replica selection, LWLC generates a list of replicas that are available in Data Grid, and it selects the replica that has the minimum T_t.

3.2.2. Replica Placement

There are several SE in the region requesting the particular rpelica, then we need select the best site to place this replica. To select the best site, LWLC creates a sorted list by the number of replica access of all SEs that requested the replica in the region. Now we will place the replica at the first SE of the above sorted list; if more than one SE have the same number of requests, LWLC algorithm selects SE which have the maximum number of different jobs that requested the particular replica. The replica is not placed at all the requested site, but only at best site in the requested region. Hence, SE utilization is very high, and can effectively improve the performance of Data Grid.

3.2.3. Replica Replacement

In the Data Grid capacity of each SE is limited, and a copy of the data also has costs. Each file in the SE has a Replica Value(RV), which is calculated as the cost for transferring a data file from another site to local site, which is calculated as the cost for transferring a data file from another site to local site next time. RV mainly considered two aspects of replica: Replica Weight (RW) and Replica Cost (RC). Larger cost will be paid if the replaced file has greater RV, so always replace the file which have the minimum RV.

RW exhibits the importance for access history in different time intervals. The RW(f) for file f is represented as:

$$RW(f)=\sum_{t=1}^{T_c} RN_t \times h^{-(T_c - t)} \quad (h \geq 2, \forall f \in F) \tag{2}$$

RW(f) indicates weight of replica f; F is the set of files that have been requested; T_c is the number of time intervals passed; t denotes the time interval t, the time interval is 1s; h is the base weight of replica; RN_t indicates the number of access for the file f at time interval t.

The time cost of transfer replica from source SE to destination SE is calculated by RC, if the deleted file has large RC, then the next request for this file will cause great time delay. On the other hand, the job always requests a deleted file which has large RC that will cause unnecessary bandwidth consumption. It mainly relates with the size of replica and effective network bandwidth. Therefore, the larger effectively network bandwidth, the shorter transfer time.

$$RC(f)=\frac{S_F(f)}{W_{rt}} \tag{3}$$

Where $S_F(f)$ is the size of desired replica f, unit is MB; W_{rt} is the effective bandwidth between source SE and destination SE, unit in MB/S.

$$RV(f)=\frac{RW(f)}{\sum_{i=1}^{n} RW(i)} \times 100\% + \frac{RC(f)}{\sum_{i=1}^{n} RC(i)} \times 100\%, \quad \forall f \in F \tag{4}$$

F is the set of files that have been requested; n represents the number of files in the SE.

When the capacity of storage element is smaller than the requested file, exit algorithm; otherwise if enough storage space exists at the best site, then replicates the selected file. If the requested file exists in the local LAN and there is not enough space in the storage element, then access the file remotely. Now, if the requested file doesn't exist in the same LAN and no

enough space in the storage element, we need to delete one or more files using the following steps:

(1) Randomly generated list datalist1 of replicas that don't exist on the access history and in the storage element. Then delete files from the above list till space is enough for the replica;

(2) If space is still not enough, then generates a list datalist2 of replicas in ascending order by RV, the replicas that are both available at the best site as well as the local LAN. Now delete files from top to bottom until there is enough space to store the new replica;

(3) If space is till insufficient, then repeat the step (2) for each LAN in current region, randomly;

(4) If space is still not enough, then generate list datalist3 of remaining replicas in ascending order by RV. Then delete files from top to bottom until there is enough space to store the new replica.

3.2.4. LWLC Algorithm

According to the ideas of Section 3.2.1 to Section 3.2.3, put forward LWLC algorithm which is improved of DHRA algorithm, it is dynamic Data Grid replication algorithm based on weight and cost of replica. Algorithm 1 describes the LWLC algorithm.

Algorithm 1: LWLC (Least weight and Least Cost)
INPUT: All files F;
OUTPUT: Best source SE, best destination SE;
(1) if(the requested file f_i is not available in the site) {//if1
(2) generate list datalist1 of f_i that are available in local LAN;
(3) select f_i from datalist1 that has minimum T_t;
(4) select the best site SE to place this replica;
(5) if(f_i's size >= SE storage size){
(6) access f_i remotely; exit;}
(7) else{//else1
(8) if(f_i's size <= SE available size) {
(9) replicate fi; exit;}
(10) else{//else2
(11) if(f_i is available in the local LAN)
(12) access f_i remotely;
(13) else{//else3
(14) Randomly generated list datalist2 of replicas that not exist on the access history and in the storage element;
(15) while(datalist2 != empty) {
(16) select a file from top of datalist2;
(17) delete this file;
(18) if(f_i's <= SE available size) {
(19) replicate f_i; exit;}
(20) }//end while
(21) generate a list datalist3 of replicas in ascending order by RV, the replicas that are both available at the best site as well as the local LAN;
(22) while(datalist3 != empty) {
(23) select a file from top of datalist3;
(24) delete this file;
(25) if(f_i's <= SE available size) {
(26) replicate fi; exit;}
(27) }//end while
(28) for(randomly selected LAN in current region) {
(29) generate a list datalist4 of replicas in ascending order by RV, the replicas that are both available at the best site as well as the local LAN;
(30) while(datalist4 != empty) {
(31) select a file from top of datalist4;
(32) delete this file;
(33) if(f_i's <= SE available size) {
(34) replicate f_i; exit;}

```
(35)            }//end while
(36)          }//end for
(37)         generate list datalist5 of remaining replicas in ascending order by RV;
(38)          while(datalist5 != empty) {
(39)            select a file from top of datalist5;
(40)            delete this file;
(41)            if(fi's <= SE available size) {
(42)              replicate fi; exit;}
(43)          }//end while
(44)        }//end else3
(45)      }//end else2
(46)    }//end else1
(47)  }//end if1
END
```

4. Experimental Results and Performance Analysis

4.1. Simulation Tool

We evaluate the performance of various replication algorithms by implementing them in OptorSim [12-13]. It is developed by the EU Data Grid project, a Java-based simulation language. Optorsim assumes that a Data Grid is composed of serveral sites, every site has zero or more Computing Elements (CEs) and Storage Elements (SEs). CE is used to execute jobs, SE store files. Resource Broker (RB) gets the user's job and dispatches each job to a proper site based on the scheduling algorithm. Each site has a Replica Manager (RM), it controls data transferring and provides a mechanism for accessing the replica catalog. Replica Optimizer (RO) within RM implements the replication algorithm.

4.2. Configuration

The topology of our simulated platform is shown in Figure 1 and this topology is from the simulated architecture of 3LHA. There are three regions: region1 consists of LAN1 and LAN2; region2 consists of LAN3, LAN4 and LAN5; region3 consists of LAN6. LAN1, LAN3 and LAN6 have three sites; other LANs all have three sites. The sites all have CE with associated SE. Site6 hold all master files at the beginning of the simulation. Intra LAN bandwidth is 1000 Mbps, Inter LAN bandwidth is 100Mbps, and Inter region bandwidth is 10Mbps. The topology parameters are shown in Table 1.

Table 1. Topology Parameters Table

Topology parameters	value
No. Of region	3
No. Of LAN	6
No. Of site	15
Size of SE	30 GB
Size of main SE	300 GB
Intra LAN bandwidth	1000Mbps
Inter LAN bandwidth	100Mbps
Inter region bandwidth	10Mbps

Table 2. Grid Job Parameters Table

Grid Job Parameters	Value
No. of job types	6
No. Of file access per job	15
Size of single file	1GB
Total size of files	100GB
Job delay	2500ms

There are 6 job types, each job type requires 15 files, the size of each file is 1GB. We assume all the files are read-only in the experimental simulation of this paper. Table 2 specifies the Grid job parameters used in our study.

4.3. Simulation Results and Discussion

We compare the performance of LWLC algorithm with three other algorithms:

(1) LRU (Least Recently used) algorithm: If the space of SE is not enough for the new replica, then delete the oldest file in the SE;

(2) LFU (Least Frequently used) algorithm: If the space of SE is not enough for the new replica, then delete the least accessed file in the SE;

(3) DHRA (Dynamic Hierarchical Replication Algorithm) algorithm: The data grid is divided into three layers. The time of waiting in the request queue is an important factor for replica selection.

In LRU and LFU replication always take place at the site where the job is executing. The files deleted by previous algorithm may be required in future and are probably available in other regions, but inter region bandwidth is lowest, so the files transferring time will increase, then the job runtime increase too.

Figure 2 shows mean job execution time based on changing number of jobs for 4 algorithms. With an increasing number of jobs, these four dynamic Data Grid replication algorithm simulation results show that the job executed time constantly increase too. The average job executed time is basic same when in small number of jobs; when the number of jobs is larger, LWLC algorithm and DHRA algorithm are able to process the jobs in the lowest mean time in comparison with other methods. But the performance of LWLC algorithm is better than the DHRA algorithm, And more obvious advantages with the increased number of jobs. In 1500 jobs the mean job time of LWLC algorithm is faster by about 25.53% compared to the DHRA algorithm; in 2100 jobs the mean job time of LWLC algorithm is faster by about 31.36% compared to the DHRA algorithm. It can be seen in the case of large number of jobs, the average job execution time of LWLC algorithm is shorter than LRU algorithm, LFU algorithm, DHRA algorithm.

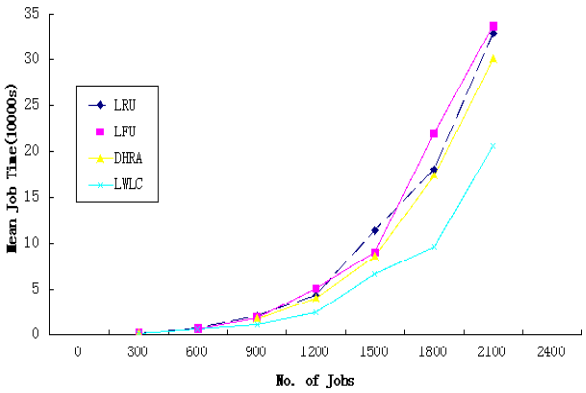

Figure 2. Mean Job Time Based on Varing Number of Jobs

5. Conclusion and Future Work

In this paper, we propose novel dynamic Data Grid replication algorithm based on weight and cost of replica, called LWLC algorithm, it improved from DHRA algorithm. In replica selection, it takes into account the number of requests that are waiting in the storage queue of source SE, the network bandwidth between source SE and destination SE also is considered. In replica placement, the replica is stored at an appropriate site, i.e. the replica in this site has the highest number of access requested, if more than one SE have the same number of requests, LWLC algorithm selects SE which has the maximum number of different jobs that requested the particular replica. In replica replacement, access history in different time interval and the time

cost of transfer replica from source SE to destination SE are introduced. The simulation result shows that LWLC algorithm can effectively improve the average jobs execution time.

In our future work, we plan to study the other aspects which affect the data grid performance: replicas differences, network congestion. Thus more effectively improve the performance of the Data Grid.

Acknowledgements

This work was financially supported by National Natural Science Foundation of China (61272124), Hebei Provincial Natural Science Foundation of China (F2012203087, S2013203002) and National Natural Science Foundation of China (61073060).

References

[1] Lili Ding, Xiaoling Wang, Wanglin Kang. Multi-Attribute Auctioning Resource in Grids: Model and Protocols. *TELKOMNIKA Indonesian Journal of Electrical Engineering.* 2013; 11(10): 5609-5616.
[2] K Ranganathan, I Foster. Identifying Dynamic Replication Strategies for a High performance Data Grid. Grid Computing-GRID 2001*. Computer Science*. 2001; 2242: 75-86.
[3] Nukarapu DT, Bin Tang, Liqiang Wang, Shiyong Lu. Data Replication in Data Intensive Scientific Applications with Performance Guarantee. *IEEE Transactions on Parallel and Distributed Systems*. 2011; 22(8): 1299-1306.
[4] Tan Cuiping, Zheng Huaiguo, Zhang Junfeng, Sun Sufen, Li Guangda. Agricultural Knowledge Grid Construction. *TELKOMNIKA Indonesian Journal of Electrical Engineering.* 2013; 11(9): 5224-5228.
[5] A Horri, R Sepahvand, Gh Dastghaibyfard. A Hierarchical Scheduling and Replication Strategy. *IJCSNS International Journal of Computer Science and Network Security*. 2008; 8(8): 30-35.
[6] Najme Mansouri, Gholam Hosein Dastghaibyfard. Job scheduling and dynamic data replication in data grid environment. *The Journal of Supercomputing*. 2013; 64(1): 204-225.
[7] Sang-Min Park, Jai-Hoon Kim, Young-Bae Ko, Won-Sik Yoon. Dynamic Data Grid Replication Strategy based on Internet Hierarchy. Grid and Cooperative Computing. *Computer Science*. 2004; 3033(2): 836-846.
[8] Ruay-shiung Chang, Jih-Sheng Chang, Shin-Yi Lin. Job scheduling and data replication on data grids. *Future Generation Computer Systems*. 2007; 23(7): 846-860.
[9] Najme Mansouri, Gholam Hosein Dastghaibyfard. A dynamic replica management strategy in data grid. *Journal of Network and Computer Application*. 2012; 35(4): 1297-1303.
[10] Najme Mansouri. An Effective weight Data Replication Strategy for Data Grid. *Australian Journal of Basic and Applied Sciences*. 2012; 6(10): 336-346.
[11] Ruay-Shiung Chang, Hui-Ping Chang, Yun-Ting Wang. A dynamic data replication strategy using access-weights in data grids. *Computer Systems and Applications.* 2008; 45(3): 277-295.
[12] The Eruopean Data Grid Project, http://eu-datagrid.web.cern.ch/eu-datagrid/
[13] Simulating data access optimization algorithms-OptorSim [EB/OL]. http://edg-wp2.web.cern.ch/edg-wp2/optimization/optorsim.html. 2004.