

A Dynamic Replication in a Data Grid using DMDR Algorithm

Lakshmi¹ Dr. Antony Selvadoss Thanamani²

¹Ph.D Research Scholar ²Assistant Professor

^{1,2}Department of Computer Science & Engineering

^{1,2}NGM College, Pollachi, India

Abstract— Data replication in data grids is an efficient technique that aims to improve response time, reduce the bandwidth consumption and maintain reliability. In this context, a lot of work is done and many strategies have been proposed. Unfortunately, most of existing replication techniques are based on single file granularity and neglect correlation among different data files. Indeed, file correlations become an increasingly important consideration for performance enhancement in data grids. In fact, the analysis of real data intensive grid applications reveals that job requests for groups of correlated files and suggests that these correlations can be exploited for improving the effectiveness of replication strategies. In this paper, we propose a new dynamic data replication strategy, called DMDR, which consider a set of files as granularity. Our strategy gathers files according to a relationship of simultaneous accesses between files by jobs and stores correlated files at the same site. In order to find out these correlations data mining field is introduced. We choose the all-confidence as correlation measure.

Keywords— replication; data grid; file correlations; data mining

I. INTRODUCTION

Data grid is a distributed collection of storage and computational resources that are not bounded within a geophysical location. It is now gaining much important interest in important areas such as high energy physics, bioinformatics, earth observations, global climate changes, image processing. One of the biggest challenges that data grids face today is to improve the data management since the techniques must scale up while addressing the autonomy, dynamicity and heterogeneity of the data sources. One way to effectively address the challenges for improving the data management is to use replication techniques. Replication creates multiple copies of the same file in several storage resources to improve response time, load balancing, reduce the bandwidth consumption and maintain reliability. In this context, many strategies have been proposed. However, most of existing replication techniques are based on single file granularity and neglect correlation among different data files. Actually, in many applications, data files may be correlated in terms of accesses and have to be considered together in order to reduce the access cost [1].

II. BACKGROUND AND RELATED WORKS

In this section, we classify and survey the replication strategies which take into account file correlations. We propose to classify these strategies according to whether they are based on data mining techniques or not in finding file correlations. The next subsection is dedicated to those strategies not based on DM techniques and the second is devoted to those using them. Due to lack of space we briefly describe the surveyed strategies.

A. Replication strategies not based on data mining techniques

1) *File clustering based replication algorithm [8]:*

The proposed algorithm group's files according to a relationship of simultaneous accesses between files and stores the replicas of the clustered files into storage nodes. This aims at satisfying the most of the expected future read accesses to the clustered files as well as processed replications for individual files being minimized under the given storage capacity limitation. The problem of clustering files is reduced to a graph partition problem where the graph is characterized by a set of vertices and a set of edges. A vertex represents a file and its size and an edge represents the number of times two files are accessed simultaneously by a single process. On the other hand, the problem of placement of groups of files is modelled as integerlinear programming. In this respect, constraints are derived from the total file size in a file cluster and the allowable storage capacity for replicas, while objectives are to minimize expected read access times to the clustered files and replication times for the individual files. The experiments on a live grid environment suggest that the proposed algorithm achieves accurate file clustering and efficient replica management.

B. Replication strategies based on data mining techniques

Data mining techniques, such as association rules, frequent sequence mining, are used mainly to identify file correlations. This section is limited to present replication strategies with data mining approaches.

1) *Replication strategy based on clustering analysis (RSCA) [15]:*

The RSCA strategy is based on the existence of correlations among the data files accessed according to the access history of users. At the first stage, a clustering analysis is conducted on the file access history by all client nodes in the grid over a period of time. The outputs of this operation are correlated file sets related to the access habits of users. At the second stage, a replication is done on the basis of those sets, which achieves his aim of pre-fetching and buffering data. The clustering method (bottom-up) adopted is used to group all the files that are similar according to a given equivalence relation into equivalence classes. The set of files in the same equivalence class are called correlative file sets. The experimental results show that RSCA is effective and available. Contrary to the other dynamic replication strategies used in the experiments, it has reduced not only the average response time of client nodes, but also the bandwidth consumption.

2) *A pre-fetching based dynamic data replication algorithm (PRA) [17]:*

The main idea of this algorithm is to make use of the characteristics that members in a virtual organization have similar interests in files to carry out a better replication optimization. The algorithm is described as following: when

a site does not have a file locally, it requests a remote site. The remote site receives the request and transfers the file to the local site. At the same time, it finds the adjacent files of the requested file by applying frequent pattern sequence mining technique on the file access sequence data base. A message containing the list of adjacent files will be sent to the local site too. At last, the local site will choose adjacent files to replication. PRA was compared with No Replication and best client strategies [18] and it is proved that it improves the average response time and the average bandwidth consumption. However a major drawback of the algorithm that it does not distinguish the different requests arriving from the different grid sites and considers them as successive file access.

3) Based on support and confidence dynamic replication algorithm in multi-tier data grid (BSCA) [24]:

The major idea of the algorithm is to pre-fetch frequently accessed files and their associated files to the location near the access site. It finds out the correlation between the data files through data access number and data access serial. It is based on support, confidence and access numbers. BSCA has two sub algorithms: data mining algorithm and replication algorithm. The data mining algorithm applied to identify frequent files is FPGrowth [25]. Moreover, support and confidence of association rules between these frequent files are computed. If the support and the confidence values between files exceed respective minimum thresholds, frequent files and their associated ones are replicated. The replication algorithm, which is applied in a decentralized manner in each node, sorts the file serial and finds out all files whose access numbers are greater than a threshold. Then, the algorithm constructs sets of related files and replicate files that do not exist in nodes. If free space is lacking, the algorithm deletes weakly correlated files. If the storage space is still insufficient, files whose access number is less than a threshold will be deleted.

III. PROPOSED REPLICATION STRATEGY

Dynamic replication is an optimization method which targets to upsurge network bandwidth and convenience of data and decrease total access time by considering different issues. The above declared issues have been talked in the suggested algorithm, that is, DMDRA (data mining based dynamic replication algorithm) for replicating files.

A. Model Creation

Data replication involves choices like when to create a replica and how many copies of a replica are mandatory. In DMDRA, replica is created if the file does not exist on the node where the request has been programmed.

B. Model Obligation

After creating a replica, the main objective is to decide where to place the replica so as to get the fast access and less access underdevelopment. The replica in DMDRA is placed on the basis of the reputation of the file and the available storage space on the node. The reputation is based on the access frequency of the file on the node.

C. Model Assortment

After assignment, the next step is to select the best replica among the pool of available replicas. The criterion for

choosing the best replica in DMDRA is based on workload of the node, availability status, available bandwidth, and computing capacity of the node.

D. Loading Space

Before the placement of replica, the amount of storage space should be taken into account. If less storage space is available, then there should be some replacement strategies involved in other algorithms. In DMDRA, the storage space is calculated in and the file having less access frequency is replaced by the replica of the file which is being requested.

E. Suppleness

The data replication strategy must be adaptive to the dynamic nature of the grid in order to provide better results. In DMDRA, if file is not available at the time of execution of job, then replica of file is created so that grid can adapt according to its dynamic nature.

To demonstrate the role of units in grid atmosphere and supposition has been made by considering the ordered structure of Internet setting used these days. It is made of many wide and local area networks joined by joining devices and swapping places. The services of Internet Service Providers (ISPs) are used by end users at different levels. There are international service providers, national service providers, regional service providers, and local service providers. At the top of the grading are the international ISPs to connect nations together [18]. Here, international ISPs are considered as master node. The national ISPs are maintained by some companies. In DMDR algorithm, the national ISPs are similar to regions, and companies can be considered as the head nodes which are responsible for the maintenance of the region. Regional ISPs are small ISPs connected to one or more national ISPs. Local ISPs provide direct service to end users. The local ISPs can be connected to regional ISP or national ISP. Local ISP is entity that just provides internet services. This is like the node in data grid where the file or the replica of the file is placed which is being requested by the user.

IV. DESCRIPTION OF DMDR ALGORITHM

The proposed DMDR also based on the network level locality. The enhanced algorithm tried to replicate files from the neighbouring region and store replica in a site where the files has been accessed frequently based on the assumption that it may require in the future.

A. DMDR Region based algorithm

Inputs: Grid Topology, Bandwidth, Data and Storage Space
Output: Find best candidate for replication, Load balancing, job execution time, No of replications, LFA (local file access), RFA (remote file access), Network usage, Network utilization, Creation of new replica.

Method:

1. *If (needed replica is not in a site)*
 {
 Search replica within the region
2. *If (replica is within the region)*
 Create list of candidate replica within the region
- Else*
3. {

```

        Search replica in other region
4. Create list of candidate replica on other region
    }
5. }
6. Fetch the most availability replica in candidate
   replica list
7. If(there is sufficient space to store the new replica)
    {
        Store it;
        Exit
8. }
9. If(replica is in the same region)
    {
        Exit;// avoid duplication in the same region
    }

```

End process

V. TESTED AND EBHR IMPLEMENTING USING MATLAB:

Job execution scenarios	values
No of jobs	100,200,300,400,500
No of job types	10
Size of single file	1 G
Scheduling algorithm	Random
Choice of execution patterns	Random Zip

Table 1: job execution scenarios

performance metrics	Description
Mean job execution time	Time to execute the job + waiting time/number of jobs completed
Effective network usage	Specifies the network utilization
Storage used(MB)	Specifies spaces used by files

Table 2: performance metrics

VI. RESULTS:

The efficiency of is calculated based Performance Metrics, that is, characteristic path length, neighbourhood connectivity and shortest path. DMDRA (data mining based dynamic replication algorithm) is compared with one other dynamic replication strategy, that is, DMDRA. On the MATLAB using the abovementioned Performance Metrics. At the time of simulation in DMDRA, there exists only one replica in a region based on the popularity of the file.

A. Shortest path (SP)

The Shortest path is calculated as time to execute a file, the time spent by a job in waiting queue divided by the number of jobs completed. It can be represented as where is the number of jobs processed by the system, is the time to execute the job, and is the waiting time of the job that has been spent in the queue. The DMDRA along with ARRA and BSCA was tested using different job numbers of jobs. The job execution time for Random Zipf Access Pattern Generator is shown in Figure 1.

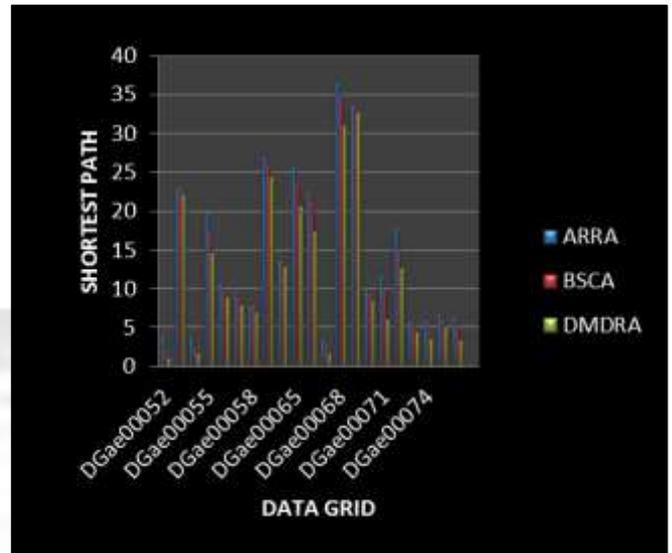


Fig. 1: Comparison of shortest path with existing algorithm

B. Characteristic path length

The characteristic path length is calculated as time to execute a file, the time spent by a job in waiting queue divided by the number of jobs completed. It can be represented as where is the number of jobs processed by the system, is the time to execute the job, and is the waiting time of the job that has been spent in the queue. The DMDRA along with ARRA and BSCA was tested using different job numbers of jobs. The job execution time for Random Zipf Access Pattern Generator is shown in Figure 2.

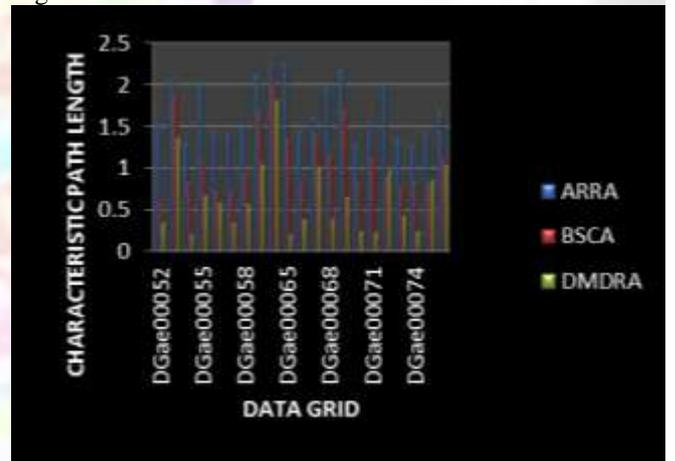


Fig. 2: Comparison of characteristic path length with existing algorithm

C. Neighbourhood connectivity

The neighbourhood connectivity is calculated as time to execute a file, the time spent by a job in waiting queue divided by the number of jobs completed. It can be represented as where is the number of jobs processed by the system, is the time to execute the job, and is the waiting time of the job that has been spent in the queue. The DMDRA along with ARRA and BSCA was tested using different job numbers of jobs. The job execution time for Random Zipf Access Pattern Generator is shown in Figure 3.

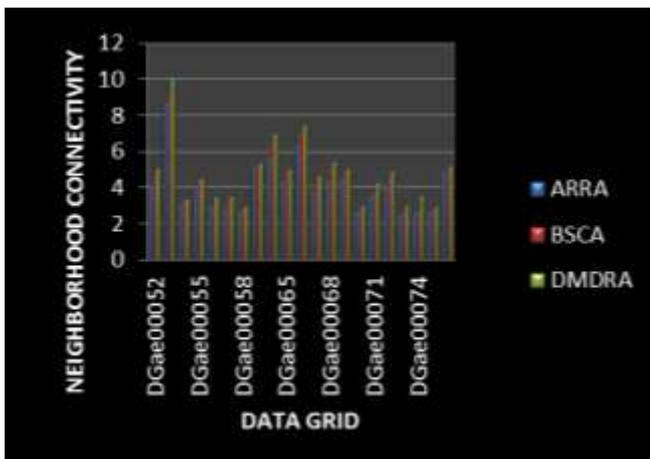


Fig. 3: Comparison of neighbourhood connectivity with existing algorithm

VII. CONCLUSION

In this paper, we have presented a DMDRA (data mining based dynamic replication algorithm) for data grids. Our strategy takes into account correlations between files and considers groups of correlated files as granularity for replication. Indeed, considering correlated files for replication improves significantly performance evaluation metrics of replication strategies, including characteristic path length, neighbourhood connectivity and shortest path. Compared with other two algorithms that are BSCA, ARRA. The experimental result shows that DMDRA gives better performance than other two algorithms. As a part of our future work, we plan to deploy our replication strategy in a real grid environment.

REFERENCES

- [1] I. Foster, C. Kesselman and S. Tuecke. 2001. The Anatomy of the Grid: Enabling Scalable Virtual Organizations, IJSA.
- [2] Kavitha Ranganathan and Ian Foster.: Identifying Dynamic Replication Strategies for a High Performance Data Grid. International Workshop on Grid Computing, Denver, November 2001.
- [3] William H. Bell, David G. Cameron, Ruben Carvajal-Schiaffino, A. Paul Millar, Kurt Stockinger, and Floriano Zini.: Evaluation of an Economy-Based File Replication Strategy for a Data Grid. In International Workshop on Agent based Cluster and Grid Computing at CCGrid 2003, Tokyo, Japan, May 2003. IEEE Computer Society Press.
- [4] Mark Carman, Floriano Zini, Luciano Serafini, and Kurt Stockinger.: Towards an Economy-Based Optimisation of File Access and Replication on a Data Grid. In International Workshop on Agent based Cluster and Grid Computing at International Symposium on Cluster Computing and the Grid (CCGrid'2002), Berlin, Germany, May 2002. IEEE Computer Society Press.
- [5] OptorSim – A Replica Optimizer Simulation: <http://edg-wp2.web.cern.ch/edgwp2/optimization/optorsim.html>
- [6] M. Tang, B.S. Lee, C.K. Yeo, X. Tang, Dynamic replication algorithms for the multi-tier data grid,

- Future Generation Computer Systems 21 (5) (2005), pp. 775– 790.
- [7] Y. Yuan, Y. Wu, G. Yang, F. Yu, Dynamic data replication based on local optimization principle in data grid, (2007).
- [8] A. Abdullah, M. Othman, H. Ibrahim, M.N. Sulaiman, A.T. Othman, Decentralized replication strategies for P2P based scientific data grid, in: Information Technology, ITSIM, International Symposium on, (2008), pp. 1–8.
- [9] Y. Ding, Y. Lu, Automatic data placement and replication in grids, in: High Performance Computing, HiPC, International Conference on, (2009), pp. 30– 39.
- [10] Neeraj Nehra, R.B. Patel, V.K. Bhat, Distributed Parallel Resource Co-Allocation with Load Balancing in Grid Computing, IJCSNS International Journal of Computer Science and Network Security, January (2007), pp. 282–291.
- [11] A. Horri, R. Sepahvand, Gh. Dastghaibiyfard, A Hierarchical Scheduling and Replication Strategy, IJCSNS International Journal of Computer Science and Network Security, August (2008), pp. 30-35.
- [12] S. M. Park, J. H. Kim, Y. B. Ko, W. S. Yoon, “Dynamic Data Replication Strategy Based on Internet Hierarchy BHR”, in: Lecture notes in Computer Science Publisher, 2004, pp. 838-846.