A Dynamic Optimal Replication Strategy in Data Grid Environment

Wuqing Zhao
School of Computer
Wuhan University
Wuhan, China
whuzhwq@163.com

Xianbin Xu
School of Computer
Wuhan University
Wuhan, China
xbxu@whu.edu.cn

Zhuowei Wang
School of Computer
Wuhan University
Wuhan, China
wangzhuowei0710@163.com

Yuping Zhang
School of Computer
Wuhan University
Wuhan, China
yuping.whu@gmail.com

Shuibing He
School of Computer
Wuhan University
Wuhan, China
hesbingxq@163.com

Abstract—**The data grid technology, which uses the scale of the Internet to provide storage capability for the mass data, has become one of the hot research topics. The high latency of the network is the bottleneck in accessing the files in data grid systems. The replication strategy can shorten the time of fetching the files by creating many replicas stored in appropriate locations. Due to the limited storage capacity of each node, it is important to have an efficient replication strategy to make sure of the overall performance. In order to solve the problem, we propose a dynamic optimal replication strategy (DORS) in this paper. First, we empirically infer a threshold from some former work to decide whether one file is needed to be copied to other nodes. Then, a measure is devised to get the importance of each file, which relies on the inner and external features of the file. Finally, the algorithm can decide which replica to replace based on the importance value. Results from the simulation show the better performance of our algorithm than other former work.**

Keywords-replication strategy; replica replacement; data grid

I. Introduction

Data Grid is the highlight in the development of the Grid technology, which can be treated as a suitable solution for high performance and data-intensive computing applications. Due to the high latency of the Wide Area Network (WAN), the main issue in Data Grid research is to design the strategy for efficient data access with low time latency. In order to solve the problem, it is a good idea to create local copies or replicas of the files in appropriate locations, and then it will be much easier and quicker for the local user to access and process the data [1]. The replicas can avoid a great deal of data transferring, and improve the efficiency of data access and the capability of fault tolerance.

The three fundamental questions any replica placement strategy has to answer are:

- When should the replicas be created?
- Which files should be replicated?
- Where should the replicas be placed?

Depending on the answers of above questions, different replication strategies have been developed. The simulation is usually used to evaluate the difference of the different strategies' performances. Since most of the datasets in the scientific data grid are read-only, the overhead of updates will not be considered in our dynamic replication strategy.

In the data grid system, it can be sure that, if there is not enough storage, a well-designed replication replacement algorithm will be needed. It can be gotten from the recent work, for example, the economy model has been used rather broadly and successfully in the data grid research [2, 3], and the replication strategy is based on the access-weight [6].

In this paper, we bring out a novel data replica replacement algorithm based on the access history, file's size and the network condition. Then the replicas which has the smallest value should be deleted, after that, the local node has enough capacity to load the new file. It is adaptive to dynamic data gird environment and can enhance the efficiency of data replica access. The good performance of our strategy is illustrated by the experiment.

The optimal replication strategy we developed is detailed in the following sections. In section 2, we compare the former related work. The theory and implementation of our strategy is discussed in section 3. In section 4, experiments are conducted to evaluate the performance of our strategy on Optorsim. In the last section, we give the conclusions.

II. Related Work

In this section, we summarize and compare the existing work for replication replacement, which includes some traditional strategies, prediction-based strategies and weight-based strategies.

In [4], Ranganathan and Foster introduced six traditional replication and caching strategies and evaluated their performance in the context of data grid. They compared the strategies by means of the access latency and bandwidth consumption of each algorithm with the simulation tool.

In [2, 3], they explored the effectiveness of economy-based resource management in data grids and proposed a policy for a data replication broker. The economic data resource management system was shown to perform with improvements when using a variable pricing scheme. This fact can be generalized to all services on a data grid that uses a common currency for transactions in a commodity market.

In [5], a decentralized architecture for adaptive media dissemination was proposed. They supposed that the popularity of the datasets satisfied the Zipf distribution. The author defined the replica weight based on levels of replica popularity. In [6], they propose a weight-based dynamic replication strategy. They collect the data access history, which contains file name, the number of requests for file, and the sources that each request came from, and then calculates the different file's weight according to their ages. According to the metric, a popular file is found and replicated to suitable sites to improve the system's performance. The weight of replica is also one of the factors in our algorithm, which is much more comprehensive than the one in [5, 6].

III. Replication Strategy

A. Scenario

When considering the requested certain files from local storage sites, the requested file list can be defined as (f_1, f_2, f_3, ⋯, f_m) ($m \geqslant 1$). For the files stored in the local storage sites, we can get it directly. In order to fetch the ones from the other sites, it is better to first transfer them to local storage in the form of replicas.

But how to decide whether the file will be copied? We need to select proper strategy to decide that, which files should be copied, and where should the replicas be placed. Another question is that the storage capacity is limited. So the replica replacement strategy is useful, if there is not enough space left for all the new files.

```
if  (Requested file in the local site)
    Do nothing;
else
    {
     if (local site has enough free space)
        Copy the file;
     else
        {
         if  (the number of the file's replicas is smaller than the threshold)
             Do nothing;
         else
             Sort the files in the local site by the values in ascending
                     order;
             Fetch the file from the sorted file list and add it into the
                     deleted list until the accumulative size of the selected
                     files is larger than or equal to the requested file;
             Delete the selected files;
        }
    }
```

Figure 1. Dynamic Optimal Replication Strategy.

Our strategy is executed from two steps. In the first step, we propose the replication strategy, which decides whether the requested file should be maintained in the local storage device, according to the number of the file's replicas. In the second step, we devise a novel replication replacement algorithm, in which we calculate the files' values in the local storage device. The file with high value will be helpful to improve the efficiency of data access, so they should be retained. It is also clear that the ones with low value will not make sense to be deleted. In figure 1, we present the replication strategy.

B. Replication decision

First, we calculate the sum of all nodes' capacity and the total size of all files in the system.

Then the storage system's relative capacity is defined as:

$$r = \frac{Q}{W} \qquad (1)$$

where Q is the sum of all nodes' capacity, W is the total size of all files in the data grid environment.

The ratio r is the threshold to decide whether the file will be copied. When the number of the file's replicas is greater than r, the file will not be copied into the local node, but when the number of the file's replicas is less than r, the file can be copied.

C. Replica Replacement Policy

After the replication decision, if the number of file's replicas is less than the threshold r, we need to copy the file to the local site. Because of the local site's limited storage capacity, it is necessary to delete some files. Which file is should be deleted? We design a novel algorithm to solve this question.

First, we calculate the access frequency of the file, noted as F(f), based on the last access history. Then we calculate the access cost, noted as C(f), which embodies the number of replicas and the maximum bandwidth with the neighbors in the system.

1) Evaluate the replica's access frequency

We propose a policy to calculate the file's access frequency, as F(f). The policy gets the information for the files from the access history in constant time interval T. Information gathered at different time intervals has different values in order to distinguish the importance between history records. In our algorithm, the value represents the quantity, and a time interval represents the time for half-life, which indicates the time required for the quantity to decay to half of the initial value [6]. Setting different value is used to evaluate the importance for history records. Older history records have smaller values, and the recent history tables are worthier than the previous access. The content is indicted below. When the file is accessed in the first time interval, the value is set 2^{-1}. At the second time interval, the value of the file becomes 2^{-2}. We can derive the value of the file at the nth time interval according the rule. The value of the file, which is accessed at the nth time interval, is depicted as the below function

$$F_n(f) = 2^{-n} \qquad (2)$$

Then we can get the file's value, F(f), which is the summation of the file's value at different time interval.

2) Calculate the replica's access cost

The cost factor is an important factor during the whole replica replacement process, which is often affected by some factors such as replica size and bandwidth. Too much bandwidth consumption may block the network and improve the possibility of fault appearance during the transfer process. Consequently, the lower the total cost is, the better the performance of replacement algorithm is.

We chose the best one--the one with the largest bandwidth, B(f), to transfer the related replica in the beginning of the transfer process. So the access cost is

$$C(f) = \frac{S(f)}{B(f)} \tag{3}$$

3) *Replica replacement policy*

Based on the above definitions, the value of the file f is defined as:

$$Value = \frac{F(f)}{N(f) * C(f)} \tag{4}$$

Combining the function 1 to function 2, we can get the below function:

$$Value = \frac{F(f) * B(f)}{N(f) * S(f)} \tag{5}$$

From the function (4), we can get every file's value in the local site. Then we sort the files in the local site by the values in ascending order, fetch the file from the sorted file list and add it into the deleted list until the accumulative size of the selected files is larger than or equal to the requested file, and delete the selected files.

IV. Performance Evaluation

To demonstrate the performance improvements of our replication strategy which is proposed in this paper, we use the Grid simulator OptorSim to evaluate the performance of our replication algorithm.

A. *Simulator and parameters*

We can achieve the realistic simulation in OptorSim. It contains a number of elements including Computing Elements (CEs), Storage Elements (SEs), Resource Broker (RB), Replica Manager (RM) and Replica Optimiser (RO). Each site consists of several CEs and SEs. The topology of our simulated platform is as in figure 2.

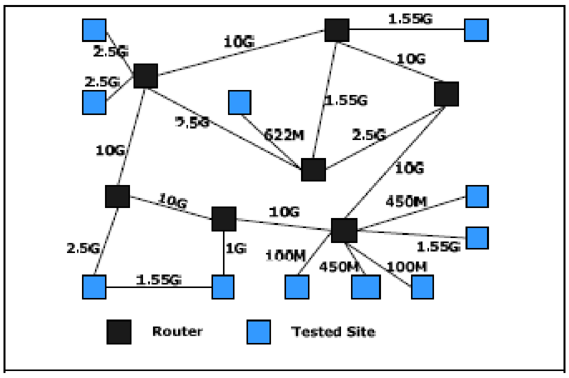

Figure 2. The topology of the simulated platform

The system workload and parameter values appear in Table 1. We assume there are 97 different files in the Grid and 2000 jobs in the Grid to measure the strategy. Each job accesses 2–50 files and each file size is initially 1 G. The storage available at an SE ranges from 30 G to 100000 G. Similar to other works in replica management for Data Grids, our simulation considers the data transmission time and assumes any internal job processing time is 100000ms. In order to simplify the requirements, we do not consider the consistency of replicas, and assume that the data is read-only in the Data Grid environment.

TABLE I. Parameters for the Data Grid

Parameter	Value
Number of storage elements	11
Number of computing elements	10
Storage element's capacity	30G---100,000G
Number of files in system	97
File size	1G
Number of jobs	2000
Number of files accessed by a job	2---50
Job processing time	100,000ms

B. *Simulation results and comparisons*

Figure 3 shows the mean job time of the three replication strategies. The job execution is about 10% faster using DORS replica algorithm than using LRU or LFU.

The number of replicas in the data grid with LFU and with LRU is more than that with the dynamic optimal replication strategy. It indicates that with the DORS we can expect less replicates which leads to its better performance than LRU and LFU.

An evaluation metric Effective Network Usage in OptorSim is used to estimate the efficiency the network resource usage, noted as E_{enu}. It is defined as:

$$E_{enu} = \frac{N_{rfa} + N_{fr}}{N_{lfa}} \tag{6}$$

Where N_{rfa} is the number of access times that CE reads a file from a remote site, N_{fr} is the total number of file replication operation, and N_{lfa} is the number of times that CE reads a file locally. A lower value indicates that the network bandwidth is used more efficiently.

Figure 4 shows the comparison of the Effective Network Usage standard of the three replication strategies. The ENU of our algorithm is lower about 15%-35% compared to the LFU or LRU algorithm. The reason is that LFU and LRU always replicate, so the large value of N_{lfa} will increase the ENU value.

In contrast, the DORS surpasses LFU and LRU algorithms in the ENU metric.

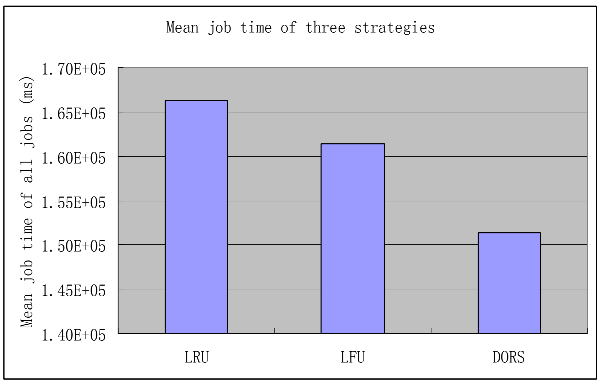

Figure 3. Mean job time of three strategies.

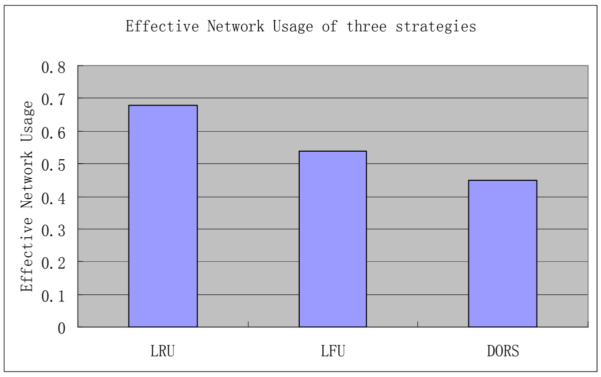

Figure 4. Effective network usage of three strategies.

Fig. 5 illustrates the effect on effective network usage as the queue length varies from 4 to 256. As the length of the job queue increases, the total network usage gradually decreases. The total network usage begins to decrease more quickly as the job queue length increases from 128 to 256. The results in this figure indicate that when the job queue length becomes longer, the DORS will trade-off the total network usage.

V. Conclusion and Future Works

In this paper, we propose an optimal replication strategy based on the file's access history and replicas' value. First, we set the threshold empirically to decide whether the file should be copied. Then, in order to overcome the problem of limited storage space in each node, we design an efficient replica replacement strategy, which is developed as a two stages process. From the simulation experiment, it can be concluded that our strategy can achieve a significant improvement of performance over former similar work. Currently, we have not tested our strategies in the real grid systems. It can be an important part of our future work. We will also try to extend our current approach with more features to enhance its performance.

References

[1] Wolfgang Hoschek , Francisco Javier Jaén-Martínez, Asad Samar, Heinz Stockinger, and Kurt Stockinger, "Data Management in an International Data Grid Project", Proceedings of the First IEEE/ACM International Workshop on Grid Computing, Springer-Verlag, 2000, pp. 77-90.

[2] 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", Proceedings of the 3rd International Workshop on Agent based Cluster and Grid Computing at International Symposium on Cluster Computing and the Grid, IEEE Computer Society, 2003, pp. 661-668.

[3] Mark Carman, Floriano Zini, Luciano Serafini, and Kurt Stockinger, "Towards an Economy-Based Optimization of File Access and Replication on a Data Grid", Proceedings of the 2nd IEEE/ACM International Symposium on Cluster Computing and the Grid, IEEE Computer Society, 2002, pp.340-345.

[4] Kavitha Ranganathan, and Ian T. Foster, "Identifying Dynamic Replication Strategies for a High Performance Data Grid", Proceedings of the Second International Workshop on Grid Computing, Springer-Verlag, 2001, pp. 75-86.

[5] Philippe Cudre-Mauroux, and Karl Aberer, "A Decentralized Architecture for Adaptive Media Dissemination", ICME'02 Proceedings, 2002, pp. 533-536.

[6] Ruay-Shiung Chang, Hui-Ping Chang, "A dynamic data replication strategy using access-weights in data grids", Journal of Supercompute, 2008.

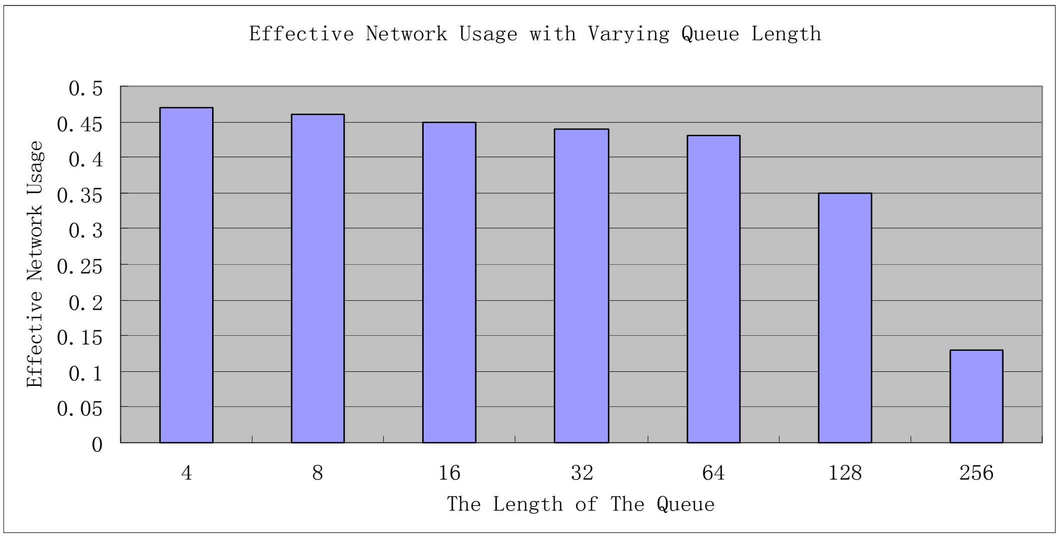

Figure 5. Effective network usage with varying queue length