Using Economic Model to Improve the Performance of BHR Dynamic Replication Algorithm

YOGITA KANESIN and GIAN CHAND SODHY
School of Computer Sciences
Universiti Sains Malaysia
11800 Penang
MALAYSIA
yogita.kanesin@gmail.com, sodhy@cs.usm.my

Abstract *- In grid computing environment, data is often replicated on many different sites to reduce access time. Static replication fixes the number of replicas whereas dynamic replication adapts to the demands of the grid by reducing or increasing the number of replicas or even moving the location of a replica to another site. Replication is highly influenced by grid users' access pattern towards a file. In this paper, the performance of the Bandwidth Hierarchy-based Replication (BHR) dynamic replication algorithm is improved by changing the way it decides to delete a file from a site. Economic model is incorporated so that a data file's future value can be calculated in order to ascertain its importance in a site. The modified BHR algorithm is tested using an available grid simulator, namely OptorSim. Other dynamic replication strategies are also simulated, and the results are compared. Our results show that the modified BHR algorithm performs better than other dynamic replication strategies.*

Keywords: Dynamic replication, data grid, grid computing.

1 Introduction

Distributed resources such as hardware and software are integrated to create a computing environment which allows multiple computer systems to share these resources. Grid computing provides this infrastructure. Terabytes of data are shared across grid environment and users accessing this data are increasing by the day. In the process of transferring these large amounts of data, the network faces congestion and latency, which consumes large amount of bandwidth. Hence data replication was implemented across grid computing environment as the remedy to this issue. Replication occurs by placing original copies of databases in various storage elements of a grid system. The copies of data (called "replicas") synchronize with other replicas in order to ensure that they store the latest updates and therefore preserving data integrity. Replication provides better reliability, accessibility and load balancing among the storage elements of a grid environment.

Dynamic and static replication strategies are currently used in grid computing environment. Static data replication only allows a predetermined number of replicas to be placed in predetermined number of locations. It does not allow customization in replication when there are changes in user behaviour. On the other hand, dynamic data replication strategy is more suitable as it adapts to changes in user behaviour and allows relocation of replicated data according to user access patterns. As it becomes evident, dynamic replication algorithms are more complicated than their static counterparts.

1.1 Replication issues

Replication is identified as the solution to overcome data availability in grid computing environment, but there are some hitches faced due to the constant increase in the number of users causing higher demand for data and files. In evaluating the effectiveness of the replication strategies, factors such as user access patterns are compared. User access pattern refers to the type of data as well as how frequently this data is being requested by the user. When there are changes in user behaviour according to their access pattern, how the number of replicas and their placement in a grid environment are determined becomes an important issue in selecting an appropriate replication algorithm.

In this paper, we try to improve the performance of a well-known algorithm, namely Bandwidth Hierarchy-based Replication (BHR), by incorporating economic model elements into it. A grid simulator is used to test our results. Comparison is made with other dynamic replication algorithms.

2 Related Work

Many dynamic replication algorithms have been proposed. We look at some of them.

Best Client Strategy [1] basically copies a file to a client if it accesses the file more than a certain number of times (i.e. exceeds a certain threshold value). However, this might lead to too many replicas.

The Cascading Replication Strategy [1] creates a replica on the same level as the site which uses the file most often. This allows other sites on the same level to access the replica.

Fast Spread Strategy [1] works by replicating a requested file along the tier before reaching the client who made the request. In other words, it creates replicas of requested file along the path to the site for future access. One drawback is that the replicas might be placed in a tier where it is least used.

Dynamic Model-Driven Replication [2] suggests replicating files on many nodes to make the file available to other nodes when the original file is not available. The placement and number of replicas need to be carefully decided.

Bandwidth Hierarchy Replication (BHR) [3] reduces the time taken to access and transfer files. It places the replicas at a high bandwidth location as well as replicating those files that are likely to be used in the near future within a region.

The Economic Model [4] replication strategy predicts whether the file brings any benefit in the future to the current site if it is replicated. A file's access frequency within a certain time interval is used to predict its future value.

3 Methodology

3.1 Framework

BHR Algorithm replicates files only when there is enough storage space. Files that exist within the same region are not replicated again. Though these factors make BHR a strong replication strategy, using the least frequently used (LFU) method for file deletion could cause files that are beneficial for future jobs to be wastefully removed. LFU basically lists all files according to their access frequency since the file was created. It keeps track of all files in the regions and how often they are being accessed throughout their existence. This does not provide enough weightage to judge a file's future access. Thus deleting files according to LFU could cause files valuable in the future to be removed as well. When the removed file is requested in the near future, it will then have to be replicated again into the storage. This might incur longer job execution time since the replication process of transferring the file back into the storage will have to take place again.

Therefore, BHR strategy can be further improved by substituting LFU with the Economic Model. The Economic Model [4] predicts a file's future value before deciding either to replicate or to remove a particular file. Economic Model has its own evaluation function that could predict a file's future value based on its past access frequency. This function is defined as the following:

$$E(f,r,n) = \sum_{i=1}^{n} p_i(f) \qquad [4]$$

The function defines the "predicted number of times a file f will be requested in the next n requests based on the past r requests in the history." [4]

The value n is derived from the function:

$$n = r\frac{\partial t'}{\partial t} \qquad [4]$$

The value n depicts the number of future requests that will be made on file f based on past r requests. The value ∂t represents the time interval when the past r requests were made and $\partial t'$ is the future time interval for the file requests to occur [4].

Economic Model predicts a file's future value based on past access frequency taken at a predefined time interval. The future file access for the similar time interval is calculated using the function shown above. This method differs from LFU as a time window is bound together with the number of requests made for the file and a probability function is used to calculate its future value during the same time interval. More values are taken into consideration in the Economic Model before deciding on the removal or replication of a file compared to the LFU method. The many factors evaluated by Economic Model will avoid the removal of important files that are actually valuable to the grid site or region. Integrating the Economic Model into BHR strategy will give an improved dynamic replication strategy. The integrated model will be identified as the BHR-Econ Model.

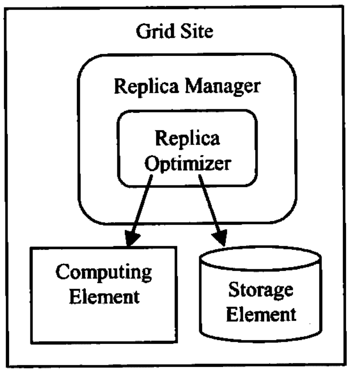

Figure 1. Grid site architecture

A grid simulator commonly identified as OptorSim [4,5,6] is used to implement the suggested framework. OptorSim is capable of providing a test bed similar to the original Grid Computing Environment by providing multiple grid sites with storage elements that can be used to create and store replicas. Users can also set the parameters of OptorSim according to their requirements to run the simulation. Different replica placement algorithms can be tested in different grid environment set-up by using OptorSim. Figure 1 (adopted from [4]) shows the architecture of OptorSim.

OptorSim allows communication between different grid sites through a Resource Broker and a Replica Manager. These allow the algorithm to access files from different grid sites. In addition to that, the existence of the Replica Optimizer allows the dynamic replication strategies to be tested as it is able to communicate with the Storage Element to decide on replica storage and removal. Hence, OptorSim is suitable to test the suggested framework.

3.2 Implementation of Framework

The suggested solution identified as BHR-Econ Model is integrated into OptorSim by placing the algorithm in the Replica Optimizer. BHR-Econ Model also has its own Storage Element function which is used when the optimizer decides to make any file deletion or replication.

BHR-Econ Model along with the other replication strategies are written in JAVA and integrated into the "optor" package of the simulator. This is where the dynamic replication strategies are normally positioned. The BHR-Econ Model is then included in the parameter file so that it can be selected to be tested using the simulator's Graphical User Interface (GUI). BHR-Econ Model Optimizer will direct the Storage Element to store replicas that are created by the optimizer as well as to remove files according to the criteria set by the BHR-Econ Model. The Storage Element will then execute the command and thus stores or removes the particular file.

The factors that are considered in evaluating the suggested framework's performance are the time taken to execute the jobs assigned as well as how best the network is utilized after the new replicas are placed on the Grid Sites. BHR-Econ Model should show the lowest job time taken to execute the jobs assigned. This will prove that the algorithm has an efficient method of placing the replicas at the site from where it can be optimized in the best form. When the results of the simulation are compared, the BHR-Econ Model should show the most minimal network utilization to prove that the replicas are placed in the best way whereby any file transaction from the replica will not heavily load the network and therefore be transferred at minimal cost.

4 Implementation and Results

4.1 Implementation Setup

The simulator, OptorSim, is executed in Linux platform and Java is used as the programming language for changing its parameters. The simulator is downloaded from OptorSim's official website [6]. OptorSim simulates a Grid Computing Environment, which consists a total of 11 grid sites. Table 1 shows the details of the grid sites within OptorSim.

Table 1. Grid sites used in OptorSim

No.	Grid Site	Computing Elements (CE)	Storage Elements (SE)	Storage Element size (GB)
1	Site 0	1	1	80
2	Site 3	1	1	33
3	Site 7	1	1	50
4	Site 8	0	1	100000
5	Site 11	1	1	63
6	Site 12	1	1	30
7	Site 13	1	1	30
8	Site 14	1	1	50
9	Site 15	1	1	50
10	Site 16	1	1	50
11	Site 17	1	1	70

The dynamic replication algorithms are placed in the Replica Optimizer which resides in the Replica Manager. Replica Optimizer handles replication of files and also the management of Storage Elements whereby it decides and handles any deletion of files due to insufficient storage space in the Storage Element to store new replicas. Thus, the suggested solution as well as other dynamic replication algorithms discussed will be placed in Replica Optimizer.

Other parameters which are used for the job processing is set to be the same for all the algorithms tested. The only factor that changes each time will be the type of optimizer selected to run the job and the number of jobs assigned. The optimizers that are tested are Cascading Replication Strategy, Fast Spread Strategy, BHR Strategy and the BHR-Econ Replication Strategy. Each optimizer has its own class of Storage Element which will be called when the optimizer decides to store or delete a file from the Storage Element.

In order to simulate using OptorSim, a few parameters are selected to set the condition on how the job processing will be carried out. These parameters remain the same throughout the testing for all the algorithms. Table 2 shows the parameters that are fixed throughout the simulation process.

Table 2. Parameters set before the start of simulation

No.	Fixed Parameters	Selected Value
1.	Scheduling algorithm used by the Resource Broker	Random
2.	Choice of access pattern generator	Sequential (files accessed in order)
3.	Initial File Distribution (Master File)	Site 8 (largest SE)

Each algorithm is tested with different number of jobs to see its efficiency as the job load increases. Job loads of 50, 200, 500, 1000 and 2000 are used for the simulation. The algorithms are tested separately for each job load. The results produced at each Grid Site are totaled for that particular algorithm. The algorithms' results are measured according to the parameters shown in Table 3.

Table 3. Performance measurements used to compare algorithms

No.	Performance Measurements	Description
1	Mean Job Time (milliseconds)	Total time taken to execute job / Number of jobs
2	Effective Network Usage	Network Traffic Load: 1 = No network traffic 0 = Network heavily loaded
3	Total Number of Replications	Total number of replicas created at the Grid Sites

Each Grid Site produces its own mean job time, effective network usage and total number of replicas created during the simulation process which will then be totaled to produce a summary table showing the performance of each algorithm when it is assigned to execute different number of jobs. The performance of the algorithm will be measured by comparing the job time taken by each algorithm to complete different loads of jobs assigned to it, the number of replicas created while processing the job load and how effectively the network is utilized during this process.

4.2 Testing and Results

Four algorithms were experimented using five different job loads in the simulator. The results derived are categorized into three different aspects in order to evaluate their performance [7].

4.2.1 Mean Job Time

The algorithms along with the suggested framework were tested using loads of 50, 200, 500, 1000 and 2000 jobs. The results can be seen in Table 4 and Figure 2. It is clear that as the job number increases, BHR-Econ Model is able to process the jobs in the lowest mean time.

Table 4. Mean job time calculated for different job loads

Number of Jobs	Mean Job Time (milliseconds)			
	BHR Strategy	BHR-Econ Model	Cascading Replication Strategy	Fast Spread Strategy
50	4688	4685	4717	4714
200	16247	15484	16460	16330
500	39424	38477	40223	39739
1000	79054	77705	80181	81223
2000	158790	156866	161107	162627

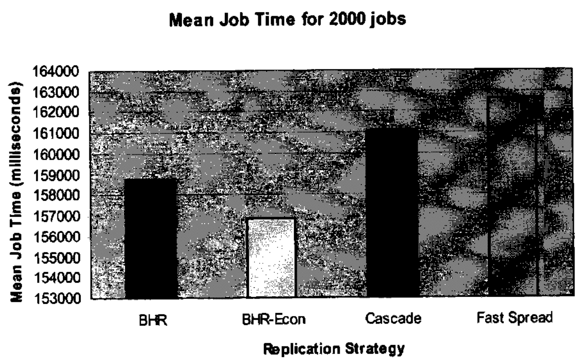

Figure 2. Mean job time taken to process 2000 jobs

The BHR-Econ Model which integrates BHR algorithms with the Economic Model file deletion method produces the lowest mean job time taken to process all the five different job loads. Hence the algorithm is able to process jobs faster than the rest of the optimizers used. BHR-Econ model applies BHR strategy in choosing the best replica to process a particular job as well as to place the replicas at the best possible site, aiming to reduce the time taken to access and transfer files from a replica. But rather than using the LFU method to delete files from Storage Element when there is no available space to store the new replica, BHR-Econ Model 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. Hence, this model reduces the chances of a replica to be deleted or replaced unless it is really worthwhile. Due to less replicas made by this algorithm, less time is taken to process the job load.

4.2.2 Number of Replicas

The algorithms are also compared by looking at the number of replications created and stored while the jobs are being processed. Table 5 and Figure 3 show the results obtained from this simulation.

Table 5. Number of replicas created for different job loads

Number of Jobs	Number of replications			
	BHR Strategy	BHR-Econ Model	Cascading Replication Strategy	Fast Spread Strategy
50	292	267	607	703
200	506	488	1885	2204
500	2137	1885	3923	4345
1000	7374	6232	8387	9223
2000	14656	13329	16275	17388

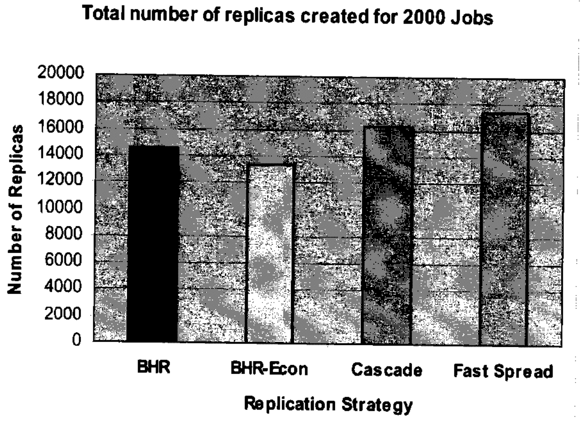

Figure 3. Number of replicas created for 2000 jobs

It is best to have the lowest number of replicas created in order to avoid any excessive time taken to produce replicas rather than using the time to process the assigned jobs. Though the replication number is low, shorter job process time depicts that the replicas are placed at the best location for them to be well optimized. In this case, the BHR-Econ Model shows the lowest number of replicas created and stored and also being able to process the jobs assigned in minimum time. This clearly indicates that the replicas are created efficiently and BHR-Econ Model has placed them at the best site from where files can be accessed and transferred at the lowest possible transfer time.

The BHR Strategy has a higher number of replications created compared to the BHR-Econ Model due to the fact that the Storage Element space is made available using LFU whenever required, hence encouraging BHR to replicate more files. By having more replicas BHR Strategy has higher mean job time as creating replicas incurs time.

4.2.3 Effective Network Usage

Network utilization is important and must be considered when it comes to transferring files from the best replica. In order to judge whether replicas are placed at the best location in a Grid Computing Environment, algorithms are compared in terms of how best the network is utilized. In OptorSim, effective network usage is represented by value 1 showing that there is no network traffic and 0 depicting that the network is heavily loaded causing longer time to access and transfer files from selected replica. An algorithm has effective network usage when replicas are placed at a location that would incur low access and transfer time. In the case of OptorSim, the values that are closer to 1 indicate efficient network usage. Table 6 and Figure 4 show the effective network usage by each algorithm.

Table 6. Network usage for different job loads

Number of Jobs	Effective Network Usage			
	BHR Strategy	BHR-Econ Model	Cascading Replication Strategy	Fast Spread Strategy
50	0.86519	0.87482	0.86382	0.49656
200	0.71274	0.81712	0.61284	0.50521
500	0.53036	0.69984	0.50078	0.47459
1000	0.49577	0.63542	0.47462	0.46084
2000	0.48377	0.53444	0.48022	0.46360

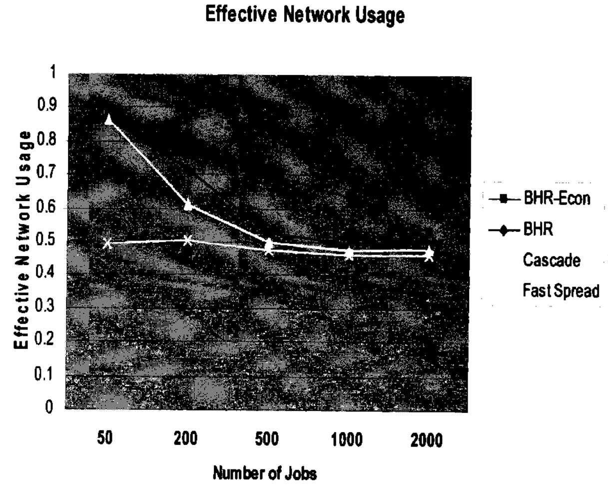

Figure 4. Network utilization by the algorithms for different number of jobs

From the graph, it can be clearly seen that as the job number increases, BHR-Econ Model has managed to maintain the most effective network usage as compared to the other algorithms. Referring back to the number of replications created by the algorithms in the previous section, BHR-Econ Model holds the lowest number of replicas produced due to replicating only those files that are worthwhile for future use and deletes those files that are of least worth. Since BHR-Econ Model is not replicating too many replicas, it is not heavily loading the network making massive file transfers and copying them into the Storage Element. Due to the mild replication, BHR-Econ Model does not require too much network usage compared to the other algorithms.

5 Summary and Future Work

5.1 Summary

In this work, various dynamic replication algorithms are studied in detail and their advantages over each other are compared. From the comparisons, a new algorithm is suggested to improve the existing replica placement algorithm. Traditional dynamic replication strategies namely the Best Client Strategy, Cascading Replication Strategy and the Fast Spread Strategy are compared with evolved replication strategies such as the BHR Strategy and Economic Model Replication Strategy. The comparisons triggered a new improved replication strategy named BHR-Econ Model. This model integrates BHR Strategy with Economic Model Strategy to derive a replication placement algorithm that is believed to perform better in Grid Computing Environment.

The suggested framework of the BHR-Econ Model is tested by integrating it into the a DataGrid Simulator known as OptorSim. It is then tested along with a few other dynamic replication strategies. Algorithms are tested on OptorSim with different job loads under the same grid environment. This means that all the other parameters are fixed except for the job number and the type of replication strategy used. This is done in order to give a wider evaluation range for the algorithms as the job load increases.

Simulation results showed that the suggested framework of BHR-Econ Model performed better than the other dynamic replication strategies tested. BHR-Econ Model showed improvement in the mean job time taken to execute the jobs assigned to it. Compared to the other algorithms, BHR-Econ took the shortest time to execute each of the different job loads. It also has the lowest number of replicas created which are optimally placed in the network since it shows minimal network utilization.

Therefore, the BHR dynamic replication strategy performs better when it is combined with the Economic Model. Together, these algorithms function in a more efficient manner in deciding the number of replicas to be created and their optimal placement in a Grid Computing Environment.

5.2 Future Work

In the current work, dynamic replication strategy is based solely on a file's access pattern. A file is selected to be stored or removed based on its access frequency. Dynamic replication placement can be improved further by integrating a few other aspects of user behaviour together rather than replicating solely based on a file's access pattern.

One aspect that can be considered in dynamic replication strategy will be to integrate cost estimation model into the replication strategy. In this case, a cost calculation can be run on the replicas and the grid sites to decide on their future placement by predicting the transfer cost that would be incurred if a replica resides on that particular grid site. By running these calculations, placement of replicas on a high transaction cost site can be predicted earlier, and hence avoided.

In addition to that, rather than just basing on a file's access pattern, dynamic replication strategy can also run a continuous check on the files from time to time to see how long a job holds on to a particular file. The longer the file is held the more important it is for the job execution. Thus files with the longest usage time can be replicated to avoid network traffic that can be caused during the long file hold-up.

These are a few suggestions that can be considered to further enhance the suggested algorithm. By integrating a few other aspects of user behaviour, replicas that are created can be more optimally utilized in the future.

References

[1] Kavitha Ranganathan and Ian Foster, "Design and Evaluation of Dynamic Replication Strategies for a High-Performance Data Grid," *Proceedings of the International Conference on Computing in High Energy and Nuclear Physics,* Beijing, September 2001.

[2] Kavitha Ranganathan, Adriana Iamnitchi, Ian Foster, "Improving Data Avail-ability through Dynamic Model-Driven Replication in Large Peer-to-Peer Communities," *Proceedings of the Workshop on Global and Peer-to-Peer Computing on Large Scale Distributed Systems,* Berlin, May 2002.

[3] Sang-Min Park, Jai-Hoon Kim, Young-Bae Ko, and Won-Sik Yoon, "Dynamic Data Grid Replication Strategy based on Internet Hierarchy," *Second International Workshop on Grid and Cooperative Computing(GCC'2003),* Shanghai, December 2003.

[4] William H.Bell, David G.Cameron, Luigi Capozza, A. Paul Millar, Kurt Stockinger, Floriano Zini, "Simulation of Dynamic Grid Replication Strategies in OptorSim," *Proc. of the ACM/IEEE Workshop on Grid Computing (Grid 2002),* Baltimore, USA, November 2002, Springer-Verlag.

[5] Houda Lamehamedi, Zujun Shentu, and Boleslaw Szymanski, "Simulation of Dynamic Data Replication Strategies in Data Grids," *Proc. 12th Heterogeneous Computing Workshop (HCW2003),* Nice, France, April 2003, IEEE Computer Science Press, Los Alamitos, CA.

[6] OptorSim – A Replica Optimizer Simulation http://edg-wep2.web.cern.ch/edgwp2/optimization/optorsim.html

[7] Yogita Kanesin, "Dynamic or Adaptive Data Replication Strategies", M.Sc. Thesis, Universiti Sains Malaysia, Penang, May 2006.