Replication in Data Grid: Determining Important Resources

Yuhanis Yusof

School of Computing, Universiti Utara Malaysia,
Malaysia
yuhanis@uum.edu.my

Abstract. Replication is an important activity in determining the availability of resources in data grid. Nevertheless, due to high computational and storage cost, having replicas for all existing resources may not be an efficient practice. Existing approach in data replication have been focusing on utilizing information on the resource itself or network capability in order to determine replication of resources. In this paper, we present the integration of three types of relationships for the mentioned purpose. The undertaken approach combines the viewpoint of user, file system and the grid itself in identifying important resource that requires replication. Experimental work has been done via OptorSim and evaluation is made based on the job execution time. Results suggested that the proposed strategy produces a better outcome compared to existing approaches.

Keywords: Grid computing, data grid, replication strategies, OptorSim.

1 Introduction

Over a number of recent years, the grid has become progressive information technology trend that enables high performance computing for scientific applications. Such a technology offers researchers the availability of powerful resources which allows them to broaden their simulations and experiments. As the grid infrastructure progress, issues are shifted towards resource management. This is realized in the Data grid where huge amount of data enables grid applications to share data files in a coordinated manner. Such an approach is seen to provide fast, reliable and transparent data access. Nevertheless, Data Grid creates a challenging problem in a grid environment because the volume of data to be shared is large despite the limited storage space and network bandwidth [1, 2]. Furthermore, resources involved are heterogeneous as they belong to different administrative domains in a distributed environment. Operationally, it is infeasible for various users to access the same data (e.g. a data file) from one single organization (e.g. site). Such situation would lead to the increase of data access latency.

Motivated by these considerations, a commonly strategy used in distributed system is also employed in Data Grid, that is replication. Experience from distributed system design shows that replication promotes high data availability, low bandwidth

H.-Y. Jeong et al. (eds.), *Advanced in Computer Science and Its Applications*, Lecture Notes in Electrical Engineering 279,
DOI: 10.1007/978-3-642-41674-3_102,

consumption, increased fault tolerance, and improved scalability. In grid environment, replication is one of the major factors affecting performance of Data Grids [3]. With this, it is suggested that well-defined replication strategies will smooth data access, and reduce job execution cost [4]. However, replication is bounded by two factors: the size of storage available at different sites within the Data Grid and the bandwidth between these sites [5]. Furthermore, the files in a Data Grid are mostly large [6, 7]; so, replication to every site and hosting unlimited number of replicas would be unfeasible. *Hence, we need to carefully decide which file that requires replication.* We propose a relationship based replication that integrates the viewpoint of three parties; user, system and grid environment. Existing study either identify the required resource based solely on users' perspective , i.e. number of access on the file [8, 9], or based on system's perspective, i.e. storage cost and read cost of a file [10-12]. As a result, there will be an insufficient utilizing of storage resource space, which in turn will lead to less storage availability. According to [13], less storage availability would lead to longer job execution time and larger network usage because only fewer replicas can be accommodated in the Data Grid, and most files will be read remotely.

The rest of this paper is structured as follows. Section 2 provides a brief description on existing work in data replication in the data grid. We include the details of our proposed replication strategy in Section 3 and the performance evaluation is presented in Section 4. Finally, we summarize the study in Section 5.

2 Background

The replication algorithm proposed in [4] determines popularity of a file by analyzing data access history. The researcher believes that the popular data in the past will remain popular in the near future. Having analyzed data access history, the average number of access, NOA, is computed. Files with NOA's value that is greater than the computer average NOA will be replicated. Hence, the order of which files to be replicated depends on the NOA. The larger the NOA, the more popular the file is and will be given a higher priority during the replication process.

Nevertheless, such an approach did not consider time period of when the files were accessed. If a file was accessed for a number of times in the past, while none was made recently, the file would still be considered popular and hence will be replicated. The algorithm proposed in [8] called Last Access Largest Weight (LALW) tries to solve this problem. The key point of LALW is to give different weights to files having different age. The LALW algorithm is similar to other algorithms [4] by means of using information on access history to determine popularity of a file. But the innovation is included by adding a tag to each access history record of a file.

The work in [14] suggested a model that helps to determine number of replicas needed to maintain the desired availability in P2P communities. With this, each site within the Data Grid is authorized to create replicas for the files. The availability of a file depends on the failure rate of peers in the network. However such a model has its own disadvantage: the exact number of replicas is not determined; rather it depends on the location service accuracy which depends on the existing number of replicas.

The accuracy of the replica location service determines the percentage of accessible files, and thus if the location service is ineffective, more replicas are created to ensure data availability. On the other hand, the work discussed in [9] proposed a replication strategy that makes replication decisions whether to increase number of replicas to face the high volume of requests, or to reduce the number of replicas to save more storage space. Evidently, increasing the number of replicas will decrease the response time, but the storage cost will be increased accordingly [9].

3 Method

In a data grid, when a resource (e.g a data file) is required by a job and is not available on a local storage, it may either be replicated or read remotely. If a file has been replicated, in the future, when it is requested, any job can accessed it quickly and the job execution time can be reduced. Due to the limited storage capacity, replication decision should be made to conform users' needs so that high demanded files (popular replicas) are efficiently maintain and files that are rarely utilized are removed. Our strategy (known as Relationship-based Replication, RBR) is designed by utilizing three types of relationships:

1) File-to-user (F2U) [15] - behavior of a file being requested by users, and notes the change to this request(whether is a growth or decay change). The relationship is represented using the exponential model. The F2U provides us with the $FileLifetime$ (FL),
2) File-to-file relationship (F2F) [15] - behavior of a file requesting other files and is noted by $FileWeight$ (FW)
3) File-to-grid (F2G) - lifetime of a file in the grid system and is represented by $File\ Age$ (FA).

Hence, the work presented in this study determines the importance of a resource (i.e data file) to three parties; users (F2U), file system (F2F) and the grid system (F2G). The integration of such information is represented by $File\ Value\ (FV)$, and is computed as using Eq. 1.

$$File\ Value(t,f) = \frac{FileLifetime(t,f)+FileWeight(t,f)}{File\ Age(t,f)} \tag{1}$$

The $FileLifetime$, $FileWeight$ and $File\ Age$ are used to compute the $File\ Value$ (FV) that is used as an indicator for the importance of a resource. The larger the value of FV, the more important the resource is, hence, requiring for replications. The operation of $FileLifetime$ and $FileWeight$ are provided in the work reported by Madi [15] and can be obtained using the followings:

$$File\ Lifetime = N_f^t \times (1 + r)$$

Where N_f^t represents the number of access for file f at time t, and r is the growth or decay rate in number of access of a file in one time interval. The value of r can be obtained using:

$$r = \left(N_f^{t+1}/N_f^t\right) - 1$$

On the other hand, the *FileWeight* is calculated as follows:

$$File\ Weight = \sum_{i=1}^{n} FL_i \times DL_i$$

where, n: total number of files in a grid system, FL: File Lifetime, and DL: dependency level of other files on the underlying file, and if there is no dependency, DL is assumed to be zero.

In addition, the age of a resource can be calculated as the time of the resource is included in the grid until the current time. Hence, it is as follows:

$$File\ Age = Time_{current} - Time_{attach}$$

4 Evaluations

In this research, the *OptorSim* [16-18] simulator was utilized to simulate the proposed RBR. The scalability of RBR is tested by the number of jobs running during the simulation. In this paper, number of jobs that is considered in our evaluation varies between 200 and 4000 jobs. The evaluation is later based on the mean job execution time (MJET)[8, 16, 17, 19-21]. We depict the obtained results in Table 1 and Figure 1 where comparison of MJET is made between the proposed RBR and four other replication strategies; LRU, LFU, LALW [8] and DRCM [15].

Table 1. Simulation Results of MJET

No. of Jobs	LRU	LFU	LALW	DRCM	RBR
200	4582	4398	3931	3792	**3545**
500	10911	8994	7839	7791	**7566**
1000	17108	17030	16241	14522	**12311**
2000	56567	55948	54133	52689	**50361**
4000	114652	106979	104129	103771	**103396**

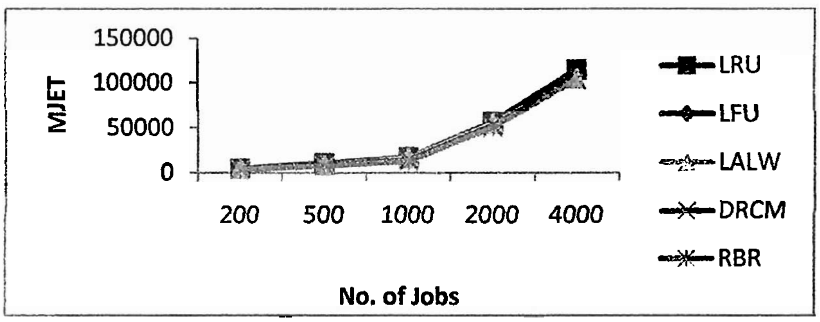

Fig. 1. Line Chart of Simulation Results

The results show a linear increase in the MJET as the number of jobs on the grid increases. This is because, as more jobs are submitted, the queue at the sites increases. If the job submission rate is higher than the grid's job processing rate, this build-up of queues is inevitable. Hence, a preferred algorithm is an algorithm that has less MJET. As shown above, the RBR is the best among existing algorithms. Utilizing the RBR, the mean job execution time is reduced and is noted to improve by 5.12% over DRCM, 24.25% over LALW, and about 7 % over LRU and LFU.

5 Conclusion

In this paper, we presented the integration of three viewpoints in identifying important resource files that requires replication. A file is assumed to be important if its access grows exponentially and is required by other files. In the simulation experiments, even though different workload has been tested, it is noted that the undertaken approach produces better result (less job execution time) as compared to its competitors. Hence, this may suggest that the proposed Relationship based Replication, RBR, could be a possible approach in data replication.

Acknowledgement. The author would like to thank Mohammed Madi for his assistant in completing the experiment. Also, author expresses the gratitude to Universiti Utara Malaysia for providing the financial and management support under the High Impact Individual research grant (s/o 12155).

References

1. Wilkinson, B.: Grid computing: techniques and applications. Chapman & Hall/CRC (2009)
2. Nicholson, C., Cameron, D.G., Doyle, A.T., Millar, A.P., Stockinger, K.: Dynamic data replication in lcg 2008. Concurrency and Computation: Practice and Experience 20, 1259–1271 (2008)
3. You, X., Chang, G., Chen, X., Tian, C., Zhu, C.: Utility-Based Replication Strategies in Data Grids. In: Fifth International Conference on Grid and Cooperative Computing, pp. 500–507 (2006)
4. Tang, M., Lee, B.S., Tang, X., Yeo, C.K.: The impact of data replication on job scheduling performance in the Data Grid. Future Generation Computer Systems 22, 254–268 (2006)
5. Venugopal, S., Buyya, R., Ramamohanarao, K.: A taxonomy of data grids for distributed data sharing, management, and processing. ACM Computing Surveys (CSUR) 38, 3 (2006)
6. Rahman, R.M., Barker, K., Alhajj, R.: Replica placement strategies in data grid. Journal of Grid Computing 6, 103–123 (2008)
7. Rahman, R.M., Barker, K., Alhajj, R.: Performance evaluation of different replica placement algorithms. International Journal of Grid and Utility Computing 1, 121–133 (2009)
8. Ruay-Shiung, C., Hui-Ping, C., Yun-Ting, W.: A dynamic weighted data replication strategy in data grids. In: AICCSA 2008: Proceedings of IEEE/ACS International Conference on Computer Systems and Applications, pp. 414–421 (2008)

9. Al Mistarihi, H.H.E., Yong, C.H.: Replica management in data grid. International Journal of Computer Science and Network Security IJCSNS 8, 22 (2008)
10. Yi-Fang, L., Pangfeng, L., Jan-Jan, W.: Optimal placement of replicas in data grid environments with locality assurance. In: 12th International Conference on Parallel and Distributed Systems, ICPADS 2006, p. 8 (2006)
11. Pangfeng, L., Jan-Jan, W.: Optimal replica placement strategy for hierarchical data grid systems. In: Sixth IEEE International Symposium on Cluster Computing and the Grid, CCGRID 2006, p. 4 (2006)
12. Mansouri, Y., Garmehi, M., Sargolzaei, M., Shadi, M.: Optimal Number of Replicas in Data Grid Environment. In: First International Conference on Distributed Framework and Applications, DFmA 2008, pp. 96–101 (2008)
13. Cameron, D.G.: Replica management and optimisation for data grids. PhD. Thesis, University of Glasgow (2005)
14. Ranganathan, K., Iamnitchi, A., Foster, I.: Improving data availability through dynamic model-driven replication in large peer-to-peer communities. In: Global and Peer-to-Peer Computing on Large Scale Distributed Systems Workshop, pp. 376–381 (2002)
15. Madi, M.: Replica Creation Algorithm for Data Grid. PhD, School of Computing, UUM College of Arts and Sciences, Universiti Utara Malaysia, Sintok (2012)
16. Cameron, D.G., Millar, A.P., Nicholson, C., Carvajal-Schiaffino, R., Stockinger, K., Zini, F.: Analysis of scheduling and replica optimisation strategies for data grids using OptorSim. Journal of Grid Computing 2, 57–69 (2004)
17. Bell, W.H., Cameron, D.G., Millar, A.P., Capozza, L., Stockinger, K., Zini, F.: Optorsim: A grid simulator for studying dynamic data replication strategies. International Journal of High Performance Computing Applications 17, 403–416 (2003)
18. Cameron, D.G., Carvajal-Schiaffino, R., Millar, A.P., Nicholson, C., Stockinger, K., Zini, F.: Evaluating scheduling and replica optimisation strategies in OptorSim. Journal of Grid Computing, 57–69 (March 2004)
19. Bell, W.H., Cameron, D.G., Capozza, L., Millar, P., Stockinger, K., Zini, F.: Simulation of Dynamic Grid Replication Strategies in OptorSim. Journal of High Performance Computing Applications 17 (2003)
20. Ben Charrada, F., Ounelli, H., Chettaoui, H.: An Efficient Replication Strategy for Dynamic Data Grids. In: Proceedings of International Conference on P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC), pp. 50–54 (2010)
21. Cameron, D.G., Carvajal-Schiaffino, R., Millar, A.P., Nicholson, C., Stockinger, K., Zini, F.: UK grid simulation with OptorSim. In: Proceedings of UK e-Science All Hands Meeting, Nottingham, UK (2003)