

REPLICATION STRATEGY BASED ON DATA RELATIONSHIP IN GRID COMPUTING

Yuhanis Yusof

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

ABSTRACT

This study discusses the utilization of three types of relationships in performing data replication. As grid computing offers the ability of sharing huge amount of resources, resource availability is an important issue to be addressed. The undertaken approach combines the viewpoint of user, system and the grid itself in ensuring resource availability. The realization of the proposed strategy is demonstrated via OptorSim and evaluation is made based on execution time, storage usage, network bandwidth and computing element usage. Results suggested that the proposed strategy produces a better outcome than an existing method even though various job workload is introduced.

KEYWORDS

Data Grid, Data Replication, Grid Computing

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 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) by combining information from users (F2U), file (F2F) and the grid (F2G) itself. The *File Value* is computed as in Eq. 1.

$$FileValue(t, f) = \frac{FileLifetime(t, f) + FileWeight(t, f)}{File\ age(t, f)}$$

The *FileLifetime*, *FileWeight* and *File Age* are used to compute the *File Value* (FV) that is used as an indicator for the volume of demand for a data file. The larger the value of FV, the more important the file is to the grid system. Hence, it will then be replicated.

The operation of *FileLifetime* and *FileWeight* are provided in the work reported by Madi[15]. If we use N_f^t to represent the number of accesses for file f at time t , and N_f^{t+1} to represent the number of accesses at time $t + 1$, the exponential growth/decay model would be given by:

$$N_f^{t+1} = N_f^t \times (1 + r)$$

where r is the growth or decay rate in number of accesses of a file in one time interval. Therefore, the value of r using the following formula can be calculated:

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

Assume t is the number of passed intervals, and N_f^t indicates the number of access for the file f at time interval t , then we get the sequence of access numbers:

$$N_f^0 N_f^1 N_f^2 N_f^3 \dots N_f^{t-1} N_f^t$$

Therefore, there are $t - 1$ time intervals, and each time interval has a growth or decay rate in number of accesses of a file. So according to the exponential growth/decay model, the equation can be written as in the following:

$$r_0 = (N_f^1/N_f^0) - 1,$$

$$r_1 = (N_f^2/N_f^1) - 1,$$

$$r_2 = (N_f^3/N_f^2) - 1,$$

$$r_{t-1} = (N_f^t/N_f^{t-1}) - 1$$

Therefore the average rate for all intervals is:

$$r = \sum_0^{t-1} r_i / t - 1$$

Having known the average accessed rate (growth or decay) for a file during the past intervals, the number of access for the upcoming time interval can be estimated, which is termed as the *File Lifetime* [15] :

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

On the other hand, the *File Weight* as described in [15] is 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.

Finally, the age of a file can be calculated as the time a file being included in the grid until the current time. Hence, it is as follows:

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

4. EXPERIMENTS

In this research, the OptorSim[16-18] simulator was utilized to simulate the proposed replication strategy. The main idea of OptorSim is when given a grid topology, resources, and a set of jobs and optimization strategy, it can simulate data movement around these job runs and supply information on various factors that could be used to evaluate the performance of the optimization strategy.

System scalability can be tested by the number of jobs running during the simulation. In this paper, to simulate different number of jobs, number of jobs that is considered in our evaluation varies between 200 and 4000 jobs. The evaluation is based on the following metrics:

Mean Job Execution Time

This is defined as the average time required executing a job starting from the time it is scheduled to the Computing Element until it complete processing all of the required files. It is calculated by accumulating the time taken by each job and divided by the number of jobs [8, 16, 17, 19-21], as shown in the following formula:

$$MJET = \frac{\sum T_{Departure} - T_{Arrive}}{n}$$

where,

T_{Arrive} : start time of job execution,

$T_{Departure}$: completion time of job execution, and

n : total number of processed jobs in the simulation.

Efficient Network Usage (ENU)

ENU is a measure of how well the replication strategy uses the network [17]. It is computed as:

$$ENU = \frac{N_{remote\ file\ access} + N_{replications}}{N_{remote\ file\ access} + N_{local\ file\ access}}$$

where $N_{remote\ file\ access}$ is the number of accesses that Computing Element reads a file from a remote site, $N_{replications}$ is the total number of file replication that occurs, and

$(N_{remote\ file\ access} + N_{local\ file\ access})$ is the number of times that Computing Element reads a file from a remote site or reads a file locally. A lower value indicates that the utilization of network bandwidth is more efficient.

Storage Element Usage

The average of all storage reserve capacity in a data grid can reflect the total system storage cost [16, 17]. The Average Storage Usage [22] metric is computed by the following equation [17]:

$$ASU = \frac{\sum_{i=1}^n \frac{U(\text{site}_i)}{C}}{N} \times 100\%$$

where,

- U : storage usage space that is reserved by the data files,
- N : number of sites in the data grid, and
- C : total capacity of the storage medium.

Computing Element Usage (CE Usage)

This is defined as the percentage of time that a CE is active (transferring or processing data) during the simulation. The CE usage of the whole grid is computed by aggregating the CE usage of each individual CE[17].

5. RESULTS

The results depicted in Table 1 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 replication strategy is a strategy that has less MJET. As shown in Table 1, for MJET, the RBR is the best among existing algorithms. Utilizing the RBR, the mean job execution time is reduced and is noted to better by 24.25% over LALW. Referring to the Average Storage Usage, by using RBR, the storage usage is reduced by outperforming LALW, 19.55%.

Table 1. : Simulation Results

Number of Jobs	Metrics	LALW	RBR
200	MJET	3931	3545
	ENU	37.87	31.92
	ASU	34.13	27.73
	CEU	22.15	23.41
500	MJET	7839	7566
	ENU	36.88	30.06
	ASU	35.71	28.78
	CEU	25.91	27.15

1000	MJET	16241	12311
	ENU	34.25	26.88
	ASU	37.12	29.97
	CEU	30.25	34.27
2000	MJET	54133	50361
	ENU	32.45	24.36
	ASU	38.63	30.54
	CEU	25.74	33.83
4000	MJET	104129	103396
	ENU	30.89	22.73
	ASU	40.11	31.54
	CEU	28.11	33.27

On the other hand, results of Efficient Network Usage (ENU) show a slight linear decrease as number of jobs on the grid increases. This is because at the start of the simulation the queues are short, but they build up while the files are replicated in the grid. Once the replication process has established, the execution time are reduced and the queue is shorten. The ENU gradually decreases with the increment in number of jobs because the amount of replication decreases over time. The RBR uses the lowest amount of network resources for the tested number of jobs because it is able to make better decision in deciding file that requires replication.

Looking at the Computing Element Usage (CEU) metric, it can be seen that the CEU generally grows as the number of jobs increases, reflecting the heavy workload. However, there is an obvious drop between 1000 and 2000, this is because with a higher number of jobs, the scheduling algorithm is sending most of the extra jobs to a few sites from where the data are easily accessible, leading to more uneven distribution of jobs around the grid. The same trend, although less marked, there is a slight drop seen with RBR. This indicates that RBR made a good balance in the grid as it is able to determine the appropriate data files that require replication. Having the required data file (replicas) at the closest site reduces the computing element usage.

6. CONCLUSION

In this paper, we presented a replication strategy that incorporates three relationships in determining the importance of a file. In the simulation experiments, different scenarios of job workload were considered in order to evaluate the proposed relationship based replication, RBR. Results showed that by integrating information on F2U, F2F and F2G, a better result was obtained as compared to an existing approach of LALW. This may suggest that the proposed strategy 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).

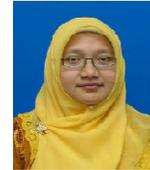
REFERENCES

- [1] B. Wilkinson, *Grid computing: techniques and applications*: Chapman & Hall/CRC, 2009.
- [2] C. Nicholson, D. G. Cameron, A. T. Doyle, A. P. Millar, and K. Stockinger, "Dynamic data replication in Icg 2008," *Concurrency and Computation: Practice and Experience*, vol. 20, pp. 1259-1271, 2008.
- [3] X. You, G. Chang, X. Chen, C. Tian, and C. Zhu, "Utility-Based Replication Strategies in Data Grids," in *Fifth International Conference on Grid and Cooperative Computing*, 2006, pp. 500-507.
- [4] M. Tang, B. S. Lee, X. Tang, and C. K. Yeo, "The impact of data replication on job scheduling performance in the Data Grid," *Future Generation Computer Systems*, vol. 22, pp. 254-268, 2006.
- [5] S. Venugopal, R. Buyya, and K. Ramamohanarao, "A taxonomy of data grids for distributed data sharing, management, and processing," *ACM Computing Surveys (CSUR)*, vol. 38, p. 3, 2006.
- [6] R. M. Rahman, K. Barker, and R. Alhaji, "Replica placement strategies in data grid," *Journal of Grid Computing*, vol. 6, pp. 103-123, 2008.
- [7] R. M. Rahman, K. Barker, and R. Alhaji, "Performance evaluation of different replica placement algorithms," *International Journal of Grid and Utility Computing*, vol. 1, pp. 121-133, 2009.
- [8] C. Ruay-Shiung, C. Hui-Ping, and W. Yun-Ting, "A dynamic weighted data replication strategy in data grids," in *AICCSA 2008: Proceedings of IEEE/ACS International Conference on computer systems and applications*, 2008, pp. 414-421.
- [9] H. H. E. Al Mistarihi and C. H. Yong, "Replica management in data grid," *International Journal of Computer Science and Network Security IJCSNS*, vol. 8, p. 22, 2008.
- [10] L. Yi-Fang, L. Pangfeng, and W. Jan-Jan, "Optimal placement of replicas in data grid environments with locality assurance," in *Parallel and Distributed Systems, 2006. ICPADS 2006. 12th International Conference on*, 2006, p. 8.
- [11] L. Pangfeng and W. Jan-Jan, "Optimal replica placement strategy for hierarchical data grid systems," in *Cluster Computing and the Grid, 2006. CCGRID 06. Sixth IEEE International Symposium on*, 2006, p. 4 pp.
- [12] Y. Mansouri, M. Garmehi, M. Sargolzaei, and M. Shadi, "Optimal Number of Replicas in Data Grid Environment," in *First International Conference on Distributed Framework and Applications, 2008. DFMA 2008.*, 2008, pp. 96-101.
- [13] David G. Cameron, "Replica management and optimisation for data grids," PhD. Thesis, University of Glasgow, 2005.
- [14] K. Ranganathan, A. Iamnitchi, and I. Foster, "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*, 2002, pp. 376-381.
- [15] M. Madi, "Replica Creation Algorithm for Data Grid," PhD, School of Computing, UUM College of Arts and Sciences, Universiti Utara Malaysia, Sintok, 2012.
- [16] D. G. Cameron, A. P. Millar, C. Nicholson, R. Carvajal-Schiaffino, K. Stockinger, and F. Zini, "Analysis of scheduling and replica optimisation strategies for data grids using OptorSim," *Journal of Grid Computing*, vol. 2, pp. 57-69, 2004.
- [17] W. H. Bell, D. G. Cameron, A. P. Millar, L. Capozza, K. Stockinger, and F. Zini, "Optorsim: A grid simulator for studying dynamic data replication strategies," *International Journal of High Performance Computing Applications*, vol. 17, pp. 403-416, 2003.
- [18] D. G. Cameron, R. Carvajal-Schiaffino, A. P. Millar, C. Nicholson, K. Stockinger, and F. Zini, "Evaluating scheduling and replica optimisation strategies in OptorSim," *Journal of Grid Computing*, pp. 57-69, March 2004.
- [19] W. H. Bell, D. G. Cameron, L. Capozza, P. Millar, K. Stockinger, and F. Zini, "Simulation of Dynamic Grid Replication Strategies in OptorSim," *Journal of High Performance Computing Applications*, vol. 17, 2003.

- [20] F. Ben Charrada, H. Ounelli, and H. Chettaoui, "An Efficient Replication Strategy for Dynamic Data Grids," in *Proceedings of International Conference on P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC)*, 2010, pp. 50-54.
- [21] D. G. Cameron, R. Carvajal-Schiaffino, A. P. Millar, C. Nicholson, K. Stockinger, and F. Zini, "UK grid simulation with OptorSim," in *Proceedings of UK e-Science All Hands Meeting*, Nottingham, UK, 2003.
- [22] S. Sivasubramanian, M. Szymaniak, G. Pierre, and M. Van Steen, "Replication for web hosting systems," *ACM Computing Surveys (CSUR)*, vol. 36, pp. 291-334, 2004.

AUTHOR

Yuhanis Yusof is a senior lecturer in School of Computing, Universiti Utara Malaysia (UUM), Malaysia. She obtained her PhD in 2007 in the area of Computer Science from Cardiff University, UK. She also holds MSc degree in Computer Science from Universiti-Sains Malaysia and Bachelor of Information Technology from UUM.



Her research interest is broadly in data analysis and management for large scale computing. This includes computational intelligence, data mining (discovering patterns of interest from data), data warehousing, information retrieval and grid computing. Her current focus is in the area of nature-inspired computing, in particularly the Artificial Bee Colony and Firefly algorithm. Such algorithms are later realized in various domains such financial analysis and grid computing.