Journal of Computer Science 4 (6): 487-493, 2008
ISSN 1549-3636

Response Time Optimization for Replica Selection Service in Data Grids

Husni Hamad E. AL-Mistarihi and Chan Huah Yong
School of Computer Sciences, University Sains Malaysia, 11800 USM, Pulau Pinang, Malaysia

Abstract: Problem Statement: Data Grid architecture provides a scalable infrastructure for grid services in order to manage data files and their corresponding replicas that were distributed across the globe. The grid services are designed to support a variety of data grid applications (jobs) and projects. Replica selection is a high-level service that chooses a replica location from among many distributed replicas with the minimum response time for the users' jobs. Estimating the response time accurately in the grid environment is not an easy task. The current systems expose high response time in selecting the required replicas because the response time is estimated by considering the data transfer time only. **Approach:** We proposed a replica selection system that selects the best replica location for the users' running jobs in a minimum response time that can be estimated by considering new factors besides the data transfer time, namely, the storage access latency and the replica requests that waiting in the storage queue. **Results:** The performance of the proposed system was compared with a similar system that exists in the literature namely, *SimpleOptimiser*. The simulation results demonstrated that our system performed better than the *SimpleOptimiser* on an average of 6%. **Conclusions:** The proposed system can select the best replica location in a lesser response time than the *SimpleOptimise.* The efficiency of the proposed system is 6% higher than the *SimpleOptimise.* The efficiency level has a high impact on the quality of service that is perceived by grid users in a data grid environment where the data files are relatively big. For example, the data files produced from the scientific applications are of the size hundreds of Terabytes.

Key words: Data grid, response time estimation, replica selection decision

INTRODUCTION

The motivation for data grid was initially driven by the data intensive applications such as scientific applications, which produce large amounts of data that need to be analyzed and shared with collaborating researchers within the scientific community who are all spread across the globe. Most of the scientific applications require accessing, storing, transferring, analyzing and replicating large amounts of data in a geographically distributed locations[4]. They face the problem of sharing the distributed data files. Indeed, scientific application domains spend a considerable effort and cost to manage the large data produced from their experiments and simulations. Furthermore, one Virtual Organization (VO) may not be able to handle the huge volume of data alone. VO involves in the combination of geographically distributed resources among different organizations or institutions such as: individuals, organizations, clusters of workstations, government bodies and businesses. Therefore, the exponential growth of scientific applications has opened up new research for computer scientists in producing an efficient techniques and algorithms for scientific applications. There are many scientific and successful grid applications which exist nowadays that motivate our research.

Most scientific applications such as: High Energy Physics (HEP)[2] and climate change modeling[8] require accessing, storing, transferring, analyzing and replicating a large amount of data in a geographically distributed locations[11]. Data replication provides a good solution for the requirement of many grid applications. Thereby, identical copies of a data file are replicated and distributed among diverse grid sites to increase data reliability and availability. Replica selection[4] is the process to select one replica location from among the many replicas based on response time. The response time is a crucial factor that influences the replica selection and thus the job turnaround time. Previous replica selection systems expose high response time. Therefore, in this study, we addressed the problem of how to estimate the response time accurately.

Corresponding Author: Husni Hamad E. AL-Mistarihi, School of Computer Sciences, Universiti Sains Malaysia, 11800 USM, Pulau Pinang

In the previous studies, the data transfer time that depends on the network bandwidth is considered to predict the response time, but the transfer time alone is not sufficient. Indeed, the storage access latency and the storage requests queue are other factors that play major roles in estimating the response time. Therefore, this study has achieved the following objectives:

- Provide an elaborate solution for estimating the response time
- Deploy our solution in a replica selection system to provide grid users with the required replicas in a minimum response time, in order to reduce the job turnaround time

Related work: In the context of the best replica selection problem, the response time is defined as the time that elapses from when the job requests the required replicas until the required replicas store in the local storage where the underlying job is being executed. Thus, the response time includes the data transfer time between the underlying two sites and the storage access latency for serving the current request. There are many studies that locate the best replica location that experience minimum response time, but the main difference among these studies is how to estimate the response time metric, because the response time can not be computed in advance[19], rather there are some criteria that play a role in estimating the response time.

The first replica selection approaches[9] aim to select the closest server to the user that houses the required replica and according to some static metric criteria such as: geographical distance in miles, topological distance in number of hops and HTTP request latency. Such approaches keep track of the last response time experienced by the client and the use of this information for future prediction. Meng Guo *et al*.[17] use probing messages which are send from servers to clients to discover the available resources and then the client uses the probing messages for deciding the best and closest server experiencing a minimum response time. However, the static metrics are not sufficient predictors for the expected response time for user requests, because the network dynamic conditions are neglected.

Dynamic replica selection approaches[8,13,19] have emerged to improve the estimation of the expected user response time, based on measurements of other network factors, such as: network bandwidth and server request latency. An intelligent prediction based on historical log files is used to decide which replica is the best and in this context, the best means the replica that has the minimum response time. These approaches depend on other grid services to monitor the resource capabilities and network status, such as the Network Weather Service (NWS)[12] and Grid Resource Information Services (GRIS). Tim *et al*.[8] have considered the network bandwidth and dynamically chooses the appropriate replica at run time. Indeed, this study adapts to dynamic changes in bandwidth. Yong *et al*.[13] have considered the GridFTP log file only as a prediction tool in order to find the replica in a minimum response time, but Sudharshan *et al*.[10] explain how the GridFTP is not sufficient for the prediction, rather a regression technique model is built for prediction on the data transfer time from the source to the sink based on three data sources: GridFTP, NWS, I/O Disk.

The authors[4,7] have considered the storage access latency with the response time. They have considered historical data information about storage latency and data transfer time as a predictor of future time, but future prediction for storage access latency is not accurate, because the grid resources-such as storage-are changed and upgraded all over the time. For example, the best replica location selected from storage X is not the best replica location after some time for the same requested site if any changes had occurred on the storage X facility. But these approaches, which depend on the historical information about the resources, can be more appropriate and applicable in a stable grid environment.

However, the storage request queue and the storage media speed were not of the previous work concerns as factors that influence the response time. In this paper, we considered these two factors for the following reasons:

- The data files are stored in a storage media which vary in speeds. Each storage media has a specific speed[15] which can be measured as an I/O data transfer rate. For example, the hard disk is faster than the tape drive and the tape drives have many types with different speeds
- Most of the storage media such as the mass storage media can serve only one request at a time and thus the other incoming requests must wait for the current request to be served. In a data grid environment, the number of requests to a high-capacity data storage device can be thousands, thus each request is queued in a storage handler queue[5]

Other approaches[1,6] use parallel download to increase the end-to-end user request time, so that the required file is downloaded from all the servers that house the underlying replica simultaneously. In such

approaches the required file is typically partitioned into segments and each segment will be downloaded from each available server. The authors in[6] proposed a new data transportation mechanism termed as rFTP that retrieves partial segment of data concurrently. The authors in[1] have proposed three techniques in retrieving the required replica, namely: Uniform technique, greedy technique and assigning with prediction technique. Uniform technique divided the required replica into equally fixed sized segments according to the available number of replicas. In greedy technique, the required replica is divided into small segments and each server is allocated one segment. In assigning with prediction technique, each server assigns a non-fixed portion of segments according to its previous performance stored in a historical data logs.

Obviously, the above mentioned approaches that uses the parallel download are feasible if and only if there are many replicas and few number of requests, but this kind of scenario has rarely happened in the reality of grids. However, the most common case of scenario often occurred when there are many requests and only a few replicas, because storages capacities and other grid resources are limited. Moreover, receiving many segments from many servers of the required file is limited in the local machine bandwidth as a result of a bottleneck.

System design: Data Grid architecture[4] as shown in Fig. 1 is divided into two levels. The upper level is a high-level of services that can make use of the lower-level of core services.

Since our proposed system is a replica selection as a high level service, some core services are used by our system as shown in Fig. 2.

The proposed system receives the users' requests from the Resource Broker (RB) and enquires the Replica Location Service (RLS) for the related physical file names and their locations. The system gets the site's related information and the network status from GRIS[12] such as: NWS, MDS and GridFTP. Accordingly, the best replica location is selected for the underlying user's job. In this context, the best replica location means the replica that has the minimum response time between the two underlying sites: the remote site that houses the replica and the local site which has the underlying job that requested the replica. Therefore, the proposed system is considered to be as a dynamic replica optimization high-level service, since the best replica location for a specific user may not be the best replica location for the same user after a time, because of the dynamicity of the grid resources. As such, the number of requests and the response time are constantly varied over time.

Our system is designed to perform caching not replication. Caching[16] is a user side phenomenon that the user decides which replica is the best and caches the required replica at the local machine. Replication is a server phenomenon that the server which houses the replicas decides which replicas is to be created and where to place these replicas.

Therefore, the proposed system is a grid service system that performs the following functions:-

- Receives the jobs from the RB
- Gathers the replica location information from RLS.
- Gathers the user previous information from the historical log file
- Gathers the current criteria values such as network bandwidth from the information service provider such as NWS, MDS and GridFTP
- Select the best replica location for grid users. In this context, the best replica location means selecting the replication site which houses the required replica and has the minimum response time
- Register new information regarding the replica transfer status into the historical log file with new updates of data

In this study, we focus on estimating the response time, which is defined as the time elapsed for moving a data file from one site to another and can be calculated by the following equation:

$$\text{Response time} = T1 + T2 + T3 \quad (1)$$

Where:
T1: Transfer time.
T2: Storage access latency.
T3: Request waiting time in the queue.

T1 represents the data transmission via a wide area network, which depends on the network bandwidth and the size of the file[16] and can be computed by the following equation:

$$T1 = \frac{\text{File Size}_{(MB)}}{\text{Bandwidth}_{(MB/SEC)}} \quad (2)$$

Typically, the operating system schedules the I/O requests in order to enhance system performance[15]. Scheduling can be implemented by maintaining a queue of requests for the storage device. Thus, the storage media speed and the number of requests in queue play a major role in the average response time experienced by

applications. Therefore, the storage access latency (T2) is the delayed time for the storage media to serve the requests and this delayed time depends on the file size and storage type. Therefore, the increase in volume of data file size causes the T2 to be increased. On another hand, the different storage media have different speeds (data transfer rate) in read and write operations. Perhaps the disk pool is faster than the tape drive and the tape drive has many types with different speeds. For example: The HP Storage Works Ultrium 920 Tape Drive speed = 120 MBps, while The HP StorageWorks Ultrium 448 Tape Drive speed = 24 MBps. Consequently, T2 can be computed by the following equation:

$$T2 = \frac{\text{File Size (MB)}}{\text{Storage Speed (MB / Sec)}} \quad (3)$$

Each storage media has many requests at the same time and the storage can serve only one request at a time. Thus, there are requests waiting in the queue. Input transfers should be done prior to an actual request; likewise, output transfers should be done after an actual write operation request. This technique is referred to as buffering[15], which balances the time required for the requests waiting in queue and the time required by the storage media for servicing the request under process. Moreover, a site is busy for the duration it transfer the required replica from the storage to the network and any other incoming data requests will have to wait for the current transaction to finish and any other requests in the queue prior to the underlying request[16]. Therefore, one has to wait for all the prior requests in the storage queue. Since the time required for the current request that is the first request in the queue is the same storage access latency time T2, the underlying request has to wait for the total T2 of the prior requests in the queue. Thus, T3 is computed by the following equation:

$$T3 = \sum_{i=1}^{n} T2 \quad (4)$$

n: Number of requests waiting in the queue prior to the underlying request

Performance evaluation: A simulation tool was needed to perform system tradeoffs to determine the performance evaluation impacts of the replica selection process. Accordingly, we conducted a thorough search on distributed and parallel systems, in particular simulation tools that support the grid features[3] such as: Bricks, GridSim, SimGrid, OptorSim, Monarc, ChicSim and MicroGrid. However, the simulation OptorSim was the most appropriate one since it simulates the data replication strategies and replica selection[2,18]. Consequently, we have considered OptorSim and made some changes to be suitable to our research.

Simulation setup: OptorSim was developed to evaluate the performance of different job scheduling and replica optimization strategies. There are a number of elements that exists in OptorSim in order to achieve realistic environment. These include: Computing Elements (CEs) to which the job is sent; Storage Elements (SEs) where data can be kept. SEs and CEs are organized in the grid. The network elements for connecting grid sites, as in reality a bandwidth among the site is represented in the simulation as well as other network status; the last two elements are the Resource Broker (RB), which submits jobs to grid sites according to some scheduling algorithms and the Replication Manager (RM) which plays a role in the replication optimization strategies. In order to simulate different replication optimization strategies, the simulation configuration should be closed to reality. OptorSim adapts the real EU DataGrid topology and configuration, the grid topology as an input to OptorSim comprises 20 sites in USA and Europe that were used during a data production form of CMS experiment[18] as shown in Fig. 3 and the other input is simulating the grid jobs and data files configuration.

CERN and FNAL are producing the original files and store them at their local storage with a capacity of 100GB each and other sites which has at least one CE and a storage capacity of 50GB each.

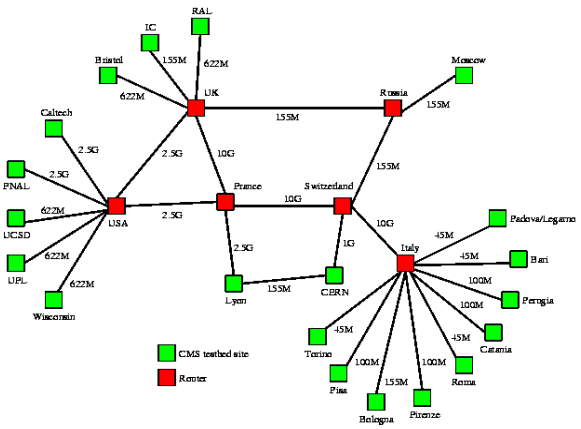

Fig. 3: Grid topology for CMS

Performance metrics: In a typical grid environment, users submit their jobs to the RB, which finds the best site to run the job in question. The jobs under execution require some data files; the optimizer finds the best locations of the required files for the jobs. However, the job will have to wait in the queue and needs time to be executed. Therefore, the job's life-time starts from the time the RB submits the job until the time the job's finished execution; and this time is called job turnaround time and includes the response time. The best replica selection according to our proposed system reduces the response time and thus reduces the job turnaround time. Therefore, the Mean Job Turnaround Time (MJTT) is suitable performance metric that evaluates our overall system performance and can be measured by the following equation:

$$MJTT = \frac{\sum_{i=1}^{n} T_{Arrive} - T_{Departure}}{n} \quad (5)$$

T_{Arriva} = The time the job arrives the system and starts execution.
$T_{Departure}$ = The time the job has finished execution.
n: = Total number of jobs processed through the system.

RESULTS AND DISCUSSION

MJTT is computed as the average of the total time required for all jobs to be executed and is measured in seconds. Since the file size and the number of jobs influence the data transfer time, we evaluated our system's performance in three different scenarios, by varying the file size and the number of jobs each time. In the first scenario, the size of the files is small, which ranged between 100 and 1000 MB. In the second scenario, the size of the files is medium, which ranged between 1 and 10 GB. In the third scenario, the size of the files is large, which ranged between 10 and 100 GB. For each scenario, three different workloads namely, 500, 1000 and 2000 jobs are experimented.

We have run the simulation for each scenario in both our system and in the SimpleOptimiser, which selects the best replica location that has minimum transfer time and already exists in OptorSim[2,18]. The SimpleOptimiser algorithm does not perform caching or replication, but rather it reads the selected replicas remotely. The results of the simulation show that the MJTT in our system is less than MJTT in SimpleOptimiser for all scenarios as shown in Table 1 and Fig. 4.

Table 1: Simulation results in different scenarios

File size	No. of Jobs	MJTT for SimpleOptimiser	MJTT for our system	Difference
Small	500	28	27	1
	1000	61	59	2
	2000	75	71	4
Medium	500	141	122	19
	1000	271	236	35
	2000	431	343	88
Large	500	789	768	21
	1000	1,519	1,480	39
	2000	2,111	2,003	108
Average		603	568	

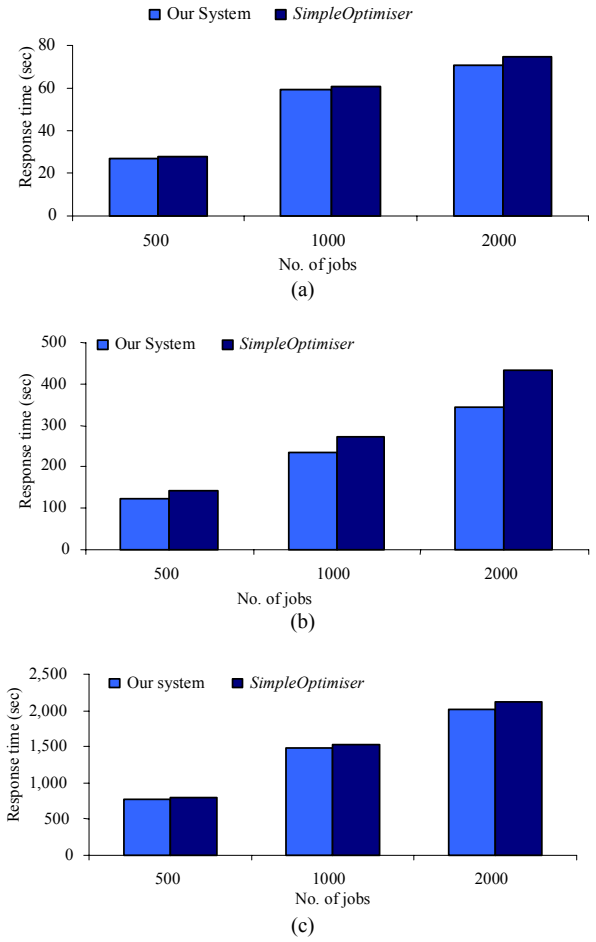

Fig. 4: Number of jobs versus response time for the two systems. (a): Small file size (b): Medium file size and (c): Large file size

The difference between the two algorithms increases when one or both of: the size of the files or the number of jobs increases, because the storage media requires more time for larger files to be serviced. The response time at our system is reduced and accordingly the job turnaround time is reduced.

Our proposed system efficiency over the SimpleOptimiser is equal to (603-568)/568×100 = 6%.

In comparison with the SimpleOptimiser, our system shows a shorter response time in all scenarios, thus indicating that our system outperforms the SimpleOptimiser. Moreover, our system is scale up to hundreds or thousands of jobs and larger size files in terms of GBs.

This study describes the replica selection service as a part of replication management services in the data grid. We have considered the response time as a criterion for selecting the best replica location for the underlying grid user. Our system can be implemented in a real grid middleware such as Globus. Finally the system can provide grid users with their required replicas in the minimum response time accurately.

ACKNOWLEDGEMENT

This study was supported by the Universiti Sains Malaysia (USM), Malaysia-Penang.

REFERENCES

1. Zhou, X.L., E. Kim, J.W. Kim and H.Y. Yeom, 2006. ReCon: A fast and reliable replica retrieval service for the data grid. In: Proceedings of the 6th IEEE International Symposium on Cluster Computing and the Grid (CCGRID'06), May 16-19, IEEE Computer Society Press, LOS ALAMITOS, USA, 1: 446-453. Doi: 10.1109/CCGRID.2006.83
2. William H. Bell1, David G. Cameron1, Ruben Carvajal-Schiaffino, A. Paul Millar, Kurt Stockinger, Floriano Zini, 2003. Evaluation of an economy-based file replication strategy for a data grid. Proceedings of the 3rd IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGRID.03), May 12-15, IEEE Computer Society, UK, pp: 661-668. Doi: 10.1109/CCGRID.2003.1199430
3. Sulistio, A., C.S. Yeo and R. Buyya, 2004. A taxonomy of computer-based simulations and its mapping to parallel and distributed systems simulation tools. Software Practice Experience, 34: 653-673. Doi: 10.1002/spe.585
4. Vazhkudai, S., S. Tuecke and I. Foster, 2001. Replica selection in the globus data grid. In: Proceedings of the 1st International Symposium on Cluster Computing and the Grid, May 15-18, Brisbane, Australia, pp: 106-113. 10.1109/CCGRID.2001. 923182
5. Park, H.J. and C.H. Lee, 2006. Sized-Based Replacement-k Replacement Policy in Data Grid Environments. Springer-Verlag Berlin, Heidelberg, pp: 353-361. ISBN: 978-3-540-68067-3, Doi: 10.1007/11946441_35
6. Jun Feng and Marty Humphrey, 2004. Eliminating replica selection using multiple replicas to accelerate data transfer on grids. In: Proceedings of the Tenth International Conference on Parallel and Distributed Systems (ICPADS'04). July 7-9. IEEE Computer Society, Los Alamitos CA. pp: 356-366. Doi: 10.1109/ICPADS.2004.1316115
7. Rahman, R.M., Ken Barker and Reda Alhajj, 2005. Replica selection in grid environment: Data-mining approach. Proceedings of the 2005 ACM symposium on Applied Computing, ACM New York, NY, USA, pp: 695- 700. Doi: 10.1145/1066677.1066836
8. Tim Ho and David Abramson, 2005. The griddles data replication service. Proceeding of the 1st International Conference on E-Science and Grid Computing (E-Science'05), Dec. 5-8, IEEE Computer Society Press, LOS ALAMITOS, USA, pp: 8. Doi: 10.1109/E-SCIENCE.2005.79
9. Mehmet Sayal, Peter Scheuermann and Radek Vingralek, 2003. Content replication in web++. In: Proceedings of the 2nd IEEE International Symposium on Network Computing and Applications (NCA'03), April 16-18, IEEE Computer Society Washington, DC, USA, Doi: 10.1109/NCA.2003.1201130
10. Sudharshan Vazhkudai and J.M. Schopf, 2003. Using regression techniques to predict large data transfers. Int. J. High Perform. Comput. Appl., 17: 249-268. Doi: 10.1177/1094342003173004
11. Chervenak, A. *et al*., 2002. Giggle: A framework for constructing scalable replica location services. In: Proceeding of the ACM/IEEE Super Computing Conference, Nov. 2002, IEEE Computer Society Press Los Alamitos, CA, USA, pp: 1-17. http://portal.acm.org/citation.cfm?id=762761.762798
12. Do-Hyeon Kim and Kyung-Woo Kang, 2006. Design and implementation of integrated information system for monitoring resources in grid computing. In: 10th International Conference on Grid Computing. Nanjing**,** China, pp: 1-6. Doi: 10.1109/CSCWD.2006.253082
13. Yong Zhao and Yu Hu, 2003. GRESS: A grid replica selection service. In: 16th International Conference on Parallel and Distributed Computing Systems (PDCS-2003), May 2006, Reno, Nevada, USA, pp: 1-6. Doi: 10.1109/CSCWD.2006.253082

14. Cameron1, D.G. *at el*., 2004. Analysis of scheduling and replica optimisation strategies for data grids using *OptorSim*. J. Grid Comput., 2: 57-69. Doi: 10.1007/s10723-004-6040-6
15. Aberham, S., G. Peter Baer and G. Greg, 2006. Operating System Principles. 7th Edn., Wiley, New York, NY, USA. ISBN 0-471-69466-5
16. Kavitha Ranganathan and Ian Foster, 2001. Identifying dynamic replication strategies for a high-performance data grid. In: Proceedings of the Second International Workshop on Grid Computing, Lecture Notes, Jan. 1, pp: 75-86. Doi: 10.1007/3-540-45644-9_8
17. Meng Guo, M.H. Ammar, E.W. Zegura and Fang Hao, 2002. A probe-based server selection protocol for differentiated service networks. IEEE Int. Conf. Commun., 4: 2353-2357. Doi: 10.1109/ICC.2002.997265
18. Bell, W.H., D.G. Cameron, L. Capozza, P. Millar, K. Stockinger and F. Zini, 2003. Optorsim-a grid simulator for studying dynamic data replication strategies. Int. J. High Perform. Comput. Appl., 17: 403-416. Doi: 10.1177/10943420030174005
19. Corina Ferdean and Mesaac Makpangou, 2003. A scalable replica selection strategy based on flexible contracts. In: Proceedings of the 3rd IEEE Workshop on Internet Applications (WIAPP'03), June 23-24, IEEE Computer Society Washington, DC, USA, pp: 95. http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=1210293