Prioritization of Replica for Replica Replacement in Data Grid

S. Pandiaraj, T. Sudalai Muthu

***Abstract*--- Datagrid provides storage facility to data grid user to store and process huge amount of data. Replication plays a major role in optimizing the Datagrid. The replacement of replica in datagrid is to be optimal to improve the performance of the Datagrid. The replacement of replica is decided based on the priority of the replica in future. The priority of the replica is based on its importance in the near future. In this paper, an approach is proposed to evaluate the priority of the replica by considering the importance of that replica in the near future. The proposed approach is evaluating the priority using the rubrics on the parameters; Frequency of access (F), Number of Access (N), and Recent access (R). The rubrics on each parameter are constructed empirically. The priority of the replica (PR) is determined by using the rubrics values and weights. The proposed approach is simulated in the OptorSim simulator. The simulation result shows that the proposed approach yielded the Hit Rate as 95% consistently performed well on various file sizes.**

***Keywords*--- Replica Replacement algorithms, Data Replications, Data Grid.**

I. INTRODUCTION

The development of computing facility follows the foot prints of the development of electricity [1]. The electrical power grid provides electrical power through a plug with appropriate wiring and devices. Similarly, the Computational Grid supplies computational powers by providing appropriate standard interfaces to which any computational device can be connected and use the resources while the source of power is completely transparent to the user.

The software employed to manage the resources and data that are used. To the user the Grid looks like one single large computer to which they can simply "plug-in" and the Grid does whatever they want it to do, automatically finding the required resources. Building a Grid is not easy, however, both in terms of the cost of hardware (CPUs, data storage devices and networks to link resources together) and developing the software to manage the resources (resource scheduling, information services, security, data management etc) [2].

The distributed nature of a Grid also brings other problems of social and political nature, when people from many countries and backgrounds have to agree on standard definitions and protocols for the Grid [3]. Despite these problems many Grids are on the verge of being used at production facilities. In the next few years it will become apparent whether the Grid will remain as specialized infrastructures for scientific experiments or be taken up by commercial interests and the general public, to become the "next generation Internet" [4,5].

II. RELATED WORK

Static replication and dynamic replication are the broad categories in the data grid environment [6]. In static replication, the replication takes place statically by pre-determined information. In dynamic replication, the replication takes place based on the status information of the data grid [7]. The dynamic replica replacement algorithms are better than static algorithms as dynamic replica replacement algorithms could optimize the replica replacement by using the information of the data grid dynamically [8].

Bsoul et al. [9] reported a Fast Spread strategies, the requested replica was first searched locally and accessed if available. If it not found, then the request travelled a short length until it was found. The requested file was replicated on all the nodes in the path for future access. If the storage was not sufficient to replicate the new replica, then the LRU and LFU were used to replace the replica with the new replica.

Bsoul et al. [9] had found out a round based algorithm based on popularity of the file. Many factors like the popularity of the file was considered to identify the appropriate file for replication at the end of each round. That algorithm yielded better performance on average file delay and bandwidth consumption.

Khanli et al. [10] proved a model called PHFS (Predictive Hierarchical Fast Spread) model to reduce the access delay. The prediction was achieved using log files and access information. The data mining techniques were applied to predict the future access of the file by applying clustering, association rule mining. Thus, PHFS established a relationship among the files for predictions in the future. The files were arranged as per the value of α in a working set called PWS (Predictive Working Set). Therefore, PHFS was able to easily choose the requested file from the working set. Thus, PHFS improved the locality of access by predicting next demand of the user.It was observed that PHFS method performed better in availability for the applications for which the work flow had a context in execution scientific projects. The PHFS was not suitable for the applications for which the requests are random in execution.

Perez et al. [11] invented a new strategy for replica management in data grid. The presented strategy focused on updating the replica dynamically hence, the consistency of the replica was maintained.

Manuscript received February 01, 2019

S. Pandiaraj, Research Scholar, Department of Computer Science and Engineering, Hindustan Institute of Technology and Science, Chennai, Tamil Nadu, India. (e-mail: sdeepandi@gmail.com)

T. Sudalai Muthu, Associate Professor, Department of Computer Science and Engineering, Hindustan Institute of Technology and Science, Chennai , Tamil Nadu, India. (e-mail: sudalaimutut@gmail.com)

Most of the replica management algorithms ignored that replica consistency by assuming that the data replica was a read only file. BRS provided better scalability on fault tolerance. As only a part of the file was replicated to the requested site, the storage was not consumed unnecessarily. It was experimented the model in a simulating environment of 50 sites data grid system. Each site had many processors and 10 storage elements with high speed Ethernet. It proved that the BRS model performed well than the HRS [11].

Sashi et al. [12] presented Dynamic Replication Algorithm (DRA) for European DataGrid. It grouped the sites into different clusters in the network topology. The sites were closely located in the cluster region. The DRA replicated the replicas within the cluster region. The most requested file was replicated in the master cluster region and it had replicated to all other cluster regions. The replication file was identified by the access frequency and popularity of the file. It considered geographical and temporal localities for identifying the popularity of the file. The DRA algorithm, experimented in Optorsim simulator. The Mean Job Execution time (MJET) was considered for performance analysis. It compared DRA with LFU and LRU. The result showed that the DRA performed better than other replacement strategies in MJET. It compared with very basic simple replacement algorithms as LRU and LFU. It is required to experiment DRA with other advanced strategies to prove the performance of the DRA [12].

Sashi et al. [13] found out an algorithm to deal with its limitations. In the model, a set of sites located closely were grouped together as network region with respect to the topological of the sites. The requested replica would be accessed fast, if the requested replica was available in the same network region. The popularity of the file was identified by using the temporal locality and the geographical localities. In that way the Modified BHR guaranteed that the replicas are present in the sites on which they get access frequently. It also experimented the Modified BHR with No Replication, LRU, LFU and BHR using OptorSim. The result showed that access cost had been reduced due to the increase rate of availability through optimization of storage space. The multitier Datagrid architecture was an efficient method for storage sharing, as well as shares the computational and network resources. It enables huge number of users to share the resources efficiently [13].

Andronikou et al. [14] had presented a QoS based algorithm which had considered the QoS parameters locality, bandwidth and network usage for replication. Higher importance replicas were replicated more number of times in the data grid. The dynamic environment of data grid is coped well by using the replica relocation manger. The replica retirement mechanism was used to handle the number of replication in storage space. The result showed that QoS model performed better in reliability than other strategies [14].

Mansouri, N. [15] investigated an algorithm to reduce the mean job execution of the grid job. A threshold value for each replica was computed using access frequency, the time taken to transfer and the size of the replication. The simulation result proved that the algorithm had better performance.

Muthu, S.et al. [16] investigated a value based replica replacement algorithm by using the importance value of each replica in the storage. The importance value was quantified by using the no. of replication, frequency of access, cost of the replica. The simulation results proved that the algorithm had better performance than others in terms of mean job time.

Muthu, S.et al. [17] reported a hybrid predictive approach to quantify the cost of replica in data grid. The algorithm was simulated in the CMS testbed environment. They have used frequency, numbers, cost of the replica to determine the cost of the replica. The replica replacement was made by replacing the least value replica.

III. PRIORITIZATION OF REPLICA

The Optimizers on each Grid site are independent agents which are responsible for providing the fastest access to data for the jobs running on the site [18]. A prediction function based on previous file access history and few assumptions of file similarity are used to determine the relative worth of files. The replica is quantified to its priority as illustrate in Table 1.

Table 1: pseudo code of Priority Evaluation of Replica

Pseudo code: Priority Evaluation of Replica
For each replica I, 1. The Rubrics value of Frequency of access (*F*), Number of Access (*N*), and Recent access (*R*) of the Replica, I, is determined from the values of Frequency of access (F_I), Number of Access (N_I), and Recent access (R_I) of the Replica, I as follows. 2. The rubrics value of Frequency is computed as $R_F = F_I * RT_F$ 3. The rubrics value of number of access is computed as $R_N = N_I * RT_N$ 4. The rubrics value of Recent access is computed as $R_R = R_I * RT_R$ 5. The weight vector $w = \{F,N,R\}$ empirically derived as $w = \{0.3, 0.3, 0.4\}$. 6. The priority of replica R_I is determined by using weight vector and rubrics values of the parameter as $P_{RI} = (0.3 * R_F) + (0.3 * R_N) + (0.4 * R_R)$ 7. The low value priority replica is replaced with the new replica.

The stored least priority value replica is replaced with the highest priority value replica to improve the performance of the data grid.

IV. EXPERIMENTAL SETUP

The proposed algorithm is simulated using Optorsim in the CMS testbed. The average of 100 independent simulations is taken as test result. The mean $\bar{x}$, the standard deviation σ and the standard error on the mean Δx of any N measurements made using repeated simulation runs had the standard forms:

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

Mean, $\bar{x} = \sum_{i=1}^{N} x_i / N$

Standard Deviation, σ $= \sqrt{\frac{\sum_{i=1}^{n}(x_i-\bar{x})^2}{N-1}}$

Standard Error, $\Delta x = \frac{\sigma}{\sqrt{N}}$

Therefore, all result calculated according to the above formulae. Each replica replacement optimization strategy was simulated on Random scheduling strategy with Random access pattern along with LRU, LFU, Binomial, Zipf and PRR replica replacements as shown in Table 2.

Table 2: Scenarios of Simulation

Job Scheduling Algorithm		File Access pattern algorithm		Replica Replacement algorithms
Random Job Scheduling	X	Random	X	LRU
				LFU
				Binomial Prediction
				Zipf Prediction
				Prioritization of Replica Approach

V. RESULTS AND DISCUSSION

The proposed algorithm has been simulated and the results are tabulated as shown in Table 3.

Table 3: Proposed Algorithm – Simulation Results

	LRU	LFU	Binomial	PR
Hit Rate(%)	92.82	94.82	96.78	96.82
MJET (ms)	13492.23	12932.39	11193.72	10284.29
ENU (%)	51.7	51.2	53.81	46.72
CEU (%)	15.8	14.9	14.1	13.2

The proposed algorithm, PRR has given better performance in Hit Rate than LRU, LFU, but it has given only ignorable amount of improvement while comparing with Binomial algorithm as shown in Figure 1.

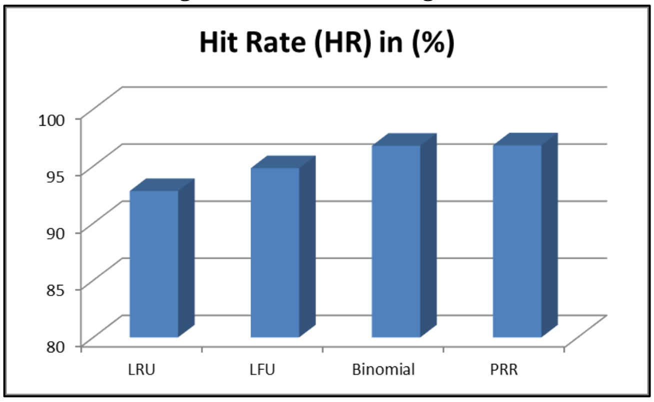

Figure 1: Hit Rate in percentile

The proposed algorithm, PRR has given better performance in MJET than LRU, LFU and Binomial as shown in Figure 2.

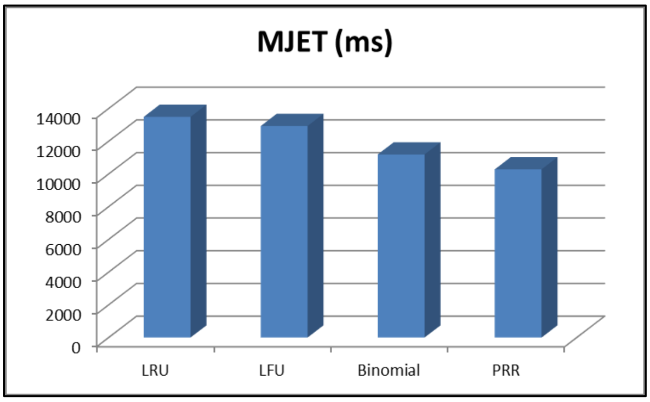

Figure 2: Mean Job Execution Time (MJET)

The proposed algorithm, PRR has given better performance in ENU than LRU, LFU and Binomial as shown in Figure 3.

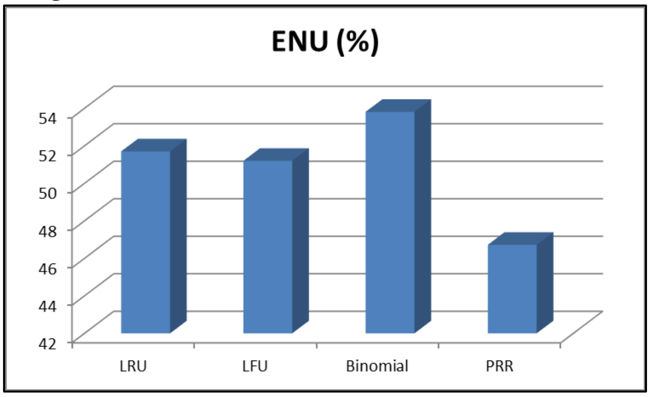

Figure 3: Effective Network Usage (ENU)

The proposed algorithm, PRR has given better performance in CEU than LRU, LFU and Binomial as shown in Figure 4.

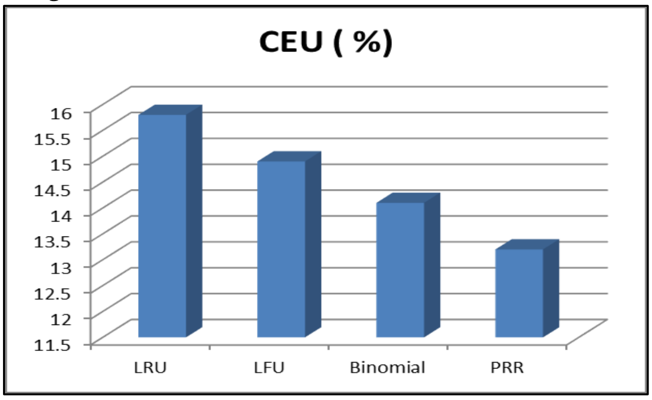

Figure 4: Computational Element Usage (CEU)

VI. CONCLUSION

The proposed algorithm has been designed by quantifying the importance of the replica by considering the frequency, number of requests, cost of the replica. The proposed algorithm is simulated in Optorsim using CMS testbed. The proposed PRR algorithm yielded better performance in Hit Rate as 4.3 %, 2.1% and 0.05% more than LRU, LFU and Binomial respectively. The proposed PRR algorithm yielded better performance in MJET as 23.77%, 20.47% and 8.12% more than LRU, LFU and Binomial respectively.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

The proposed PRR algorithm yielded better performance in ENU as 9.63%, 8.75% and 13.17% more than LRU, LFU and Binomial respectively. The proposed QwR algorithm yielded better performance in CEU as 16.45%, 11.4% and 6.38% more than LRU, LFU and Binomial respectively. The results evidenced that the proposed algorithm, PRR has well performed than LRU and LFU, but PRR algorithm has given ignorable amount of improvement over the Binomial algorithm.

REFERENCES

1. T. S. Muthu and K. R. Kumar, "A Log-based predictive approach for replica replacement in data grid," 2017 4th International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, 2017, pp 1-6.
2. Ranjana P, George, “A Improving network capacity for effective traffic management using graphs”, International Journal of Applied Engineering Research, vol. 10, no.7, pp. 16853-16863, 2015.
3. Sathish N, Ranjana P, “Secure remote access fleet entry management system using UHF band RFID”, Advances in Intelligent Systems and Computing, vol. 216, pp. 141-149, 2014.
4. Tos, Uras, Riad Mokadem, Abdelkader, Hameurlain, Tolga Ayav, and Sebnem Bora, "Dynamic replication strategies in data grid systems: a survey," The Journal of Supercomputing, vol. 71, no. 11, pp. 4116-4140 , 2015.
5. T. Amjad, "A survey of dynamic replication strategies for improving data availability in data grids," Future Generation Computer Systems, vol. 28, no.2, pp. 337-349, February, 2012.
6. T. S. Muthu, R. Vadivel, A. Ramesh and G. Vasanth, "A novel protocol for secure data storage in Data Grid environment," Trendz in Information Sciences & Computing(TISC2010), Chennai, 2010, pp. 125-130.
7. T. Sudalai Muthu and R. Masillamani, "A novel protocol for secure data storage in Data Grid," 2010 First International Conference On Parallel, Distributed and Grid Computing (PDGC 2010), Solan, 2010, pp. 184-189.
8. S. Nazanin and M. Amir, “PDDRA: A new pre-fetching based dynamic data replication algorithm in data grids,” International Journal of Future Generation Computer Systems, vol. 28, no. 1, pp. 666-681, 2012.
9. M. Bsoul, A. Al-Khasawneh, E. Eddien Abdallah, and Y. Kilan, “Enhanced fast spread replication strategy for data grid,” Journal of Network and computer applications, vol. 12, no. 4, pp. 113-121, 2010.
10. Leyli Mohammad Khanli, Ayaz Isazadeh, Tahmuras N. Shishavan, “PHFS: A dynamic replication method, to decrease access latency in the multi-tierdata grid”, Future Generation Computer Systems, vol. 1, no. 27, pp. 233-244, 2011.
11. J. M. Pérez, Félix García-Carballeira, Jesús Carretero, Alejandro Calderón, Javier Fernández, “Branch replication scheme: A new model for data replication in large scale data grids,” Journal Future Generation Computer Systems, vol. 26, no.1, pp.12-20, 2010.
12. K. Sashi and A.S. Thanamani, “A new dynamic replication algorithm for European data grid,” Third Annual ACM Bangalore Conference, pp. 17-25, 2010.
13. K. Sashi and A. S. Thanamani, "Dynamic replication in a data grid using a Modified BHR Region Based Algorithm," Future Generation Computer Systems, vol. 27, pp. 202-210, 2011.
14. V. Andronikou, K. Mamouras, K. Tserpes, D. Kyriazis, T. Varvarigou, “Dynamic QoS-aware data replication in grid environments based on data importance,” Future Gener. Comput. Syst, vol. 28, pp. 544–553, 2012.
15. N. Mansouri, "A Threshold-based dynamic data replication and parallel job scheduling strategy to enhance datagrid," Journal of Cluster Computing, vol. 17, no. 3, pp. 957-977, 2014.
16. T. SudalaiMuthu and K. RameshKumar, “A Value based dynamic replica replacement strategy in datagrid,” International Journal of Control Theory and Applications, vol. 10, no. 26, pp. 448-462, 2017.
17. T. SudalaiMuthu and K. RameshKumar, "Hybrid predictive approach for replica replacement in data grid," 2017 International Conference on Computer Communication and Informatics (ICCCI), Coimbatore, 2017.
18. T. SudalaiMuthu, K. RameshKumar and K. Sarukesi, “A Weight based replica replacement algorithm in datagrid”, Journal of Advanced Research in Dynamical and Control Systems, vol. 6, pp. 1164-1178, 2017.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication