

## Replica Selection Algorithm in Data Grids: The Best-Fit Approach

Ayman Jaradat<sup>1\*</sup>

<sup>1</sup> Department of Computer Science and Information, College Of Science Al-Zulfi, Majmaah University, Majmaah 11932, Saudi Arabia

\* Corresponding author's e-mail: [ay.jaradat@mu.edu.sa](mailto:ay.jaradat@mu.edu.sa)

### ABSTRACT

The design of Data Grids allows grid facilities to manage data files and their corresponding replicas from all around the globe. Replica selection in Data Grids is a complex service that selects the best replica place amongst several scattered places based on quality of service parameters. All replica selection algorithms look for the best replica for the requesting users without taking into account the limitation of their network or hardware capabilities to find the best fit. This leaves capable users with limited ability to connect with the best replica places without fully utilizing their download speed. It furthermore compromises the best replica places and shifts capable users to lower quality replica places and degrades the whole Data Grid environment. To improve quality of service parameters the solution we propose is, a matching algorithm that matches the capabilities of grid user with replica providers that are the best fit. This best-fit approach takes into account both the capabilities of grid users and the capabilities of replica places and creates matches of almost similar capabilities. Simulation results proved that the best-fit algorithm outperforms previous replica selection algorithms.

**Keywords:** replica selection, data grid; grid computing.

### INTRODUCTION

Data Grids are a great mechanism for problem solving in virtual organizations [1]. The emergence of Grid Computing was crucial facilitating groundwork for many disciplines such as engineering, science, earth sciences, high energy physics, astronomy and molecular biology. Grid computing has the possibility to back different types of applications. For example, data-intensive applications, computer-intensive applications and the applications demanding scattered services. Three sorts of Grids advanced to provision these applications. They are characterized as Service Grids, Computational Grids and Data Grids. Data grids are anticipated to be the solution to the huge data storage issue and computational power issue of numerous current scientific projects. The evolving move in scientific applications in various fields like climate simulation [2], high-energy physics and data mining, demonstrates that such applications manipulate

and yield enormous amounts of data [3]. These rustling huge data needs to be put in storage for additional exploration and shared with scholars collaborating within the scientific community who are scattered everywhere in the world.

Replica Selection [4] is a mechanism to choose the best replica place amongst several replica locations according to quality of service (QoS) parameters. There are several QoS parameters like response time (RsT), security, availability, reliability, and cost are very important and have crucial impact on the Grid environment as explained in [1, 5–7]. RsT or for simplicity time is a vital element that effects the replica selection and thus the job turnaround time. Earlier replica selection algorithms addressed time QoS parameter as the only parameter, and dedicated or shed on estimating it and put all the efforts in selecting the replica place with fastest replica movement, from source to sink.

However, selecting the best replica place to a user who is not capable of utilizing the full speed

replica transfer due to limitation of his network or hardware capabilities compared to the selected location, prevent other capable users from fully utilizing it, and switches them to a poorer quality grid sites. As a result, this situation demonstrates the bad impact on the capable users' performance and this will impact the whole data grid environment. Therefore, in this research, the replica selection is addressed from both sides, the sender and the receiver. It is, in all situations, better to choose a replica location that is consistent with or similar to the receivers capabilities. The previous mentioned case converts the problem from selecting the best replica location to selecting the best-fit replica location in order to improve the whole grid environment. So, the proposed algorithm takes into account two variables to decide the best replica location first it considers the capability of the user and second the capability of the data grid site. This algorithm is titled The best-fit algorithm (BFA). In this research, suitable replica and best-fit replica will be used interchangeably.

The rest of the paper is organized as follows. Section II describes the related work in replica management. Section III presents the system design while Section IV shows the performance evaluation. Section V presents the results and discussion. Finally the conclusion and future work are given.

## RELATED WORK

The challenge in replica selection research began by considering the RsT as the only QoS factor to address. RsT is the amount of time required to transfer the replica from source to sink's local storage where the running task is being executed [8]. In this context many researches addressed the problem of finding the best replica place that show minimum RsT. However; since the RsT cannot be computed in advanced the main challenge was how to estimate the RsT [9] as there are many factors that play a role in estimating RsT. These include maintaining the previous RsT histories experienced by grid users.

Early replica selection paradigms [10] intended to choose the closest replica location to the grid user based on some static metric factors like : topological distance based hop counts, geographical distance in miles and utilize it for future prediction. On the other hand, authors of [11] utilize probing messages that sent from replica sites to grid user to first check the availability of the machine that

holds the replica and second to find the grid site that shows shortest RsT to conclude it as the best replica location. Nevertheless, these approaches ignored the dynamic network nature which make these static metrics not adequate estimators to anticipate RsT. Authors of [8] argued that RsT is the sum of storage access latency (SAL), transfer time and request waiting time in the queue.

Dynamic paradigms for selecting the best replica [2, 9, 12] have arisen to enhance the estimated RsT anticipated by the data-grid-users, according to calculations of the network criteria, like: the hosted server latency for the request and the bandwidth of the network. A smart forecast founded on historic system logs is utilized to select the best replica location that shows the slightest transfer time. These techniques relay on data grid services to observe the resource powers and the conditions of the network, like the Grid Resource Information Services (GRIS) and the Network Weather Service (NWS) [13]. Authors of [2] during the run time, have used the bandwidth of the networks to automatically decide on the proper site that holds the replica. Definitely, this approach adjusts based on bandwidth fluctuations. On the other hand, the estimation tool of the researchers of [12] have relayed on only logs originated form GridFTP. However authors of [14] clarified why the GridFTP is not adequate for RsT estimation, instead a regression method has been constructed to estimate the time to move the best replica to the required location utilizing grid services : I/O Disk, NWS, GridFTP. Moreover researchers of [4,7] have incorporated the RsT with SAL where storage latency past history and data transfer time are utilized as a forecaster of future SAL but future forecast for SAL cannot be very precise due to grid resources dynamicity like storages as they fluctuate or upgrade as time passes.

For instance, the best replica place chosen from a given storage will not be the best place after a period of time due to utilization by the same or other grid user. However, techniques rely on historic information are more applicable in a steady grid environment. On the other hand authors of [8] take into account two new parameters firstly storage media specification which differ in speeds from one type to other or from brand to brand [15] the speed is measured as an I/O data transfer rate. It is well known that the tape drive is slower than the hard disk and the hard disks or the tape drives have different types

or speeds. Secondly, taking into account orders in que as utmost mass storage devices can't take more than one order simultaneously as a result the incoming orders have to wait in ques before being served. In fact the thousands of orders to a certain storage device in grid environment, so the orders are queued in a storage handler queue which effect the RsT [16].

Furthermore, strategies [17, 18] utilized parallel replica movement to escalate RsT where the needed replica is transferred concurrently from all the grid sites that held the required replica. In these techniques the needed replica is divided into parts and each part is moved from the available data grid site. The researchers of [18] introduced a novel replica transference technique labelled as rFTP that fetches replica fragments simultaneously while authors of [17] introduced three fetching approaches that are: matching with prediction approach, greedy approach and uniform approach. In matching with prediction approach, every replica site is in charge of a non-fixed quota of fragments that are consistent with its past performance saved in the log files. On the other hand, in the greedy approach, the needed data file is divided into fragments and each data grid site is assigned one fragment while in the uniform approach the needed replica is divided into equally fixed sized fragments consistent with the available number of data grid sites. However, concurrent approaches are suitable when there are few replica orders and lots of replica sites but usually the vice versa happened huge number of orders and a limited replica locations.

Finally, as mentioned above several researches integrated a number of QoS parameters a part of RsT such as reliability, security, availability in the selection process [1, 19–24]. In addition to these QoS parameters some works also added users preferences to guide the selection process [6, 25]. Moreover, a recent work went for group decision making by considering multiple users preferences simultaneously prior to assign users to replica grid sites [6].

## SYSTEM DESIGN

The structure of data grids is built into two levels lower and upper. The lower one provides fundamental services and the upper consists of a high-level service that supply fundamental services in the lower level. The new proposed algorithm is amongst the high level services which utilize several fundamental services. BFA works by obtaining the data grid users order from the Resource Broker (RB) and questions the Replica Location Service (RLS) for the associated physical replica title and their places. BFA receives the data grid nodes interrelated situation and the status of the network from GRIS [13] like GridFTP, Monitoring and Discovery Service (MDS) and NWS. Consequently, the best-fit place is chosen for the grid user's task. In fact, the best-fit replica place in this research means the place with specifications that can send the file to the grid user in a speed that the user can absorb without delay or without a congestion at the user site. Figure 1, presents an overview of BFA and its related entities.

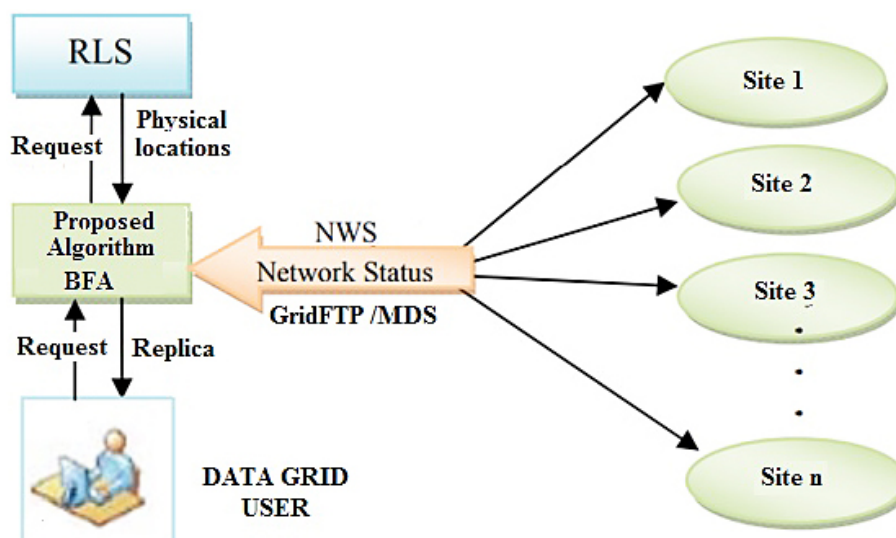


Fig. 1. Overview of BFA & Surrounded Entities

Hence, BFA is a high level, dynamic , optimization service, where the best-fit replica place for a certain data grid user may not be the best-fit replica place for other grid users or the same user as time passes and this is because the dynamic nature of data grid resources. To select the best-fit replica place the algorithm behaves as follows:

1. Obtain the tasks from the RB.
2. Collects the places of the replicas from RLS.
3. Collects historical logs from the system files.
4. Collects the instant values of QoS parameters like the bandwidth from the information service provider like GridFTP, MDS and NWS.
5. Evaluate each replica place based on its RsT and assign it a rate value.
6. Evaluate the requesting user QoS parameters and assign it a RsT value.
7. Sort replica places based on their RsT values in an ascending order.
8. Select the best-fit replica place for the underlying user, which shows a RsT of equal or greater value.
9. Record the latest information concerning the data transmission speed into the historic logs.

RsT in this study is the total of three parameters that are: Transfer time (TT), Storage access latency (SAL) and the waiting time in the queue (WTQ) and is calculated by the following equation:

$$RsT = TT + SAL + WTQ \tag{1}$$

TT denotes the replica movement through the network, that rely on the network bandwidth and the file size [16] obtained by the formula:

$$TT = \frac{\text{File Size MB}}{\text{Bandwidth}} \text{ (MB /SEC)} \tag{2}$$

The important role of the operating system is to schedule the I/O requests in a manner that enhance the system performance [26].

Scheduling considers queued requests for the storage device. Therefore, the number of requests in queue and the speed of the storage device and have a significant influence in the average RsT.

As a result, SAL is time delay required by the storage device to reply to an order and it is relevant to the file size and storage speed. Consequently, larger replicas yields higher SAL which can be calculated by the following formula:

$$SAL = \frac{\text{File Size MB}}{\text{Storage Speed}} \text{ (MB /SEC)} \tag{3}$$

Typically, orders arrive to each storage device that can't serve them simultaneously as it can serve one order at a time, hence several orders queued forming a waiting in the queue. Therefore, the current order has to wait for all former queued orders. The time needed for the underlying order that is the first one in the queue of order is the same SAL time, so has to wait for the sum of SALs of the preceding orders in the queue. Accordingly, WTQ is calculated by the following formula:

$$WTQ = \sum_{i=1}^n SAL \tag{4}$$

where: n – number of orders waiting in the queue preceding the current order.

## PERFORMANCE EVALUATION

A collection of simulation tools available to support the data grids structures [27]. For example MicroGrid, ChicSim, Monarc, OptorSim, SimGrid and Bricks. But, after carrying techniques [3, 28]. Accordingly, this research adopted out a comprehensive search on parallel and distributed algorithms simulation tools the conclusion is OptorSim simulator is the most suitable one for the proposed algorithm as it mainly focuses on replica choice tactics and data replication OptorSim by performing tiny amendments to make it more fit to the proposed algorithm.

### Simulation Setup

In order to assess the performance of replica selection methods the OptorSim simulator was built and was unlike other task scheduling algorithms. OptorSim consists of several elements to attain realistic grid environment such as Computing Elements (CEs) where tasks are directed, Storage Elements (SEs) where data is saved. The network elements to link data grid nodes. Like real grids the bandwidth between the grid nodes is embodied in the simulation. The other elements are the Resource Broker (RB), that assigns tasks to grid nodes based on the scheduling technique. Where Replication Manager (RM) contributes to the replication optimization techniques. To be consistent with real grids, OptorSim imitate the real EU DataGrid configuration and topology. The topology includes 20 grid nodes in Europe and

the USA that have been utilized during the data production form of CMS experiment [28]. FNAL and CERN sites create the huge data and save them locally with a storage volume of 100 GB each and other grid nodes have a minimum of one CE with a storage volume of 50 GB each.

**Performance metrics**

The roles of grid users takes place by sending their tasks to RB that searches for the best grid location to carry out the task. However, usually these tasks need data files so the optimizer role is to get the best place of the needed files for the tasks. Nevertheless, the task needs to wait in the queue and requires some time to be accomplished.

As a result, the task’s duration begins when the RB sends the task and completed once the task’s accomplished. This duration is known as task turnaround time and comprises the RsT. The best-fit replica selection based on the new proposed algorithm decreases the RsT and accordingly decreases the task turnaround time. For that reason, the Mean task Turnaround Time (MTTT) is an appropriate performance metric that assesses the performance of the proposed algorithm and is calculated by the following formula:

$$MTTT = \sum_{1}^n CNTM - ARTM / n \quad (5)$$

where: arrival-time (ARTM) – Task’s arrival and execution time; completion-time (CNTM) – Task’s completion time; n – the over-all number of tasks.

**RESULTS AND DISCUSSION**

MTTT is the proposed metric to evaluate and compare the new algorithm and it stands for the average of the whole times needed to complete all the tasks submitted to the grid. Based on the fact that the sizes of the replicas and the number of the tasks (workload) impact the replicas transfer time, the performance of the new algorithm has been evaluated through nine different cases, by changing the size of the replicas and the number of the tasks each time. In the first case, small replicas has been used with sizes range between 100 to 1000 MB and workload 600 tasks while the second case with a workload of 1200 tasks and the third case with a workload equal to 1800 tasks as shown in Table 1. Cases four to six are similar to one to three but with medium replicas with sizes range between 1 to 10 GB as shown in Table 2. Cases seven to nine are similar to the previous ones but with large replicas with sizes range between 10 to 100 GB as shown in Table 3. Each case has been experimented ten times each time the replica sizes and sites QoS are varied randomly. The experiments carried out utilizing the proposed algorithm (BFA) and the previous algorithm (PA) in order to choose the best replica place that has the lowest transferal time and already included in OptorSim [2, 18]. The algorithms do not make replication or caching, instead they read the chosen replicas remotely. The simulation results demonstrated that the MTTT experienced by the proposed algorithm is less than MTTT experienced by the previous algorithm for all cases as presented in Tables 1

**Table 1.** Simulation results for small replicas

Experiment No	600 Tasks			1200 Tasks			1800 Tasks		
	BFA	BFA	Efficiency	BFA	BFA	Efficiency	BFA	BFA	Efficiency
1	56.68	49.34	14.9%	557.39	488.37	14.1%	2889.05	2506.02	15.3%
2	54.25	47.31	14.7%	555.56	487.61	13.9%	2895.28	2525.93	14.6%
3	55.49	48.1	15.2%	551.1	479.11	15.0%	2721.57	2413.30	12.8%
4	55.42	48.12	15.2%	539.98	473.38	14.1%	2755.83	2413.58	14.2%
5	55.58	48.22	15.3%	550.23	482.91	13.9%	2799.91	2431.64	15.1%
6	57.6	50.21	14.7%	546.6	477.10	14.6%	2874.45	2513.58	14.4%
7	56.22	49.2	14.3%	574.43	496.99	15.6%	2909.35	2524.46	15.2%
8	54.71	47.29	15.7%	563.15	488.15	15.4%	2802.35	2442.28	14.7%
9	53.25	46.59	14.3%	539.89	473.01	14.1%	2799.13	2438.91	14.8%
10	55.73	48.2	15.6%	558.44	487.30	14.6%	2796.01	2441.66	14.5%
MTTT	55.49	48.26	15.0%	553.68	483.39	14.5%	2824.30	2465.14	14.6%

**Table 2.** Simulation results for medium replicas

Experiment No	600 Tasks			1200 Tasks			1800 Tasks		
	PA	BFA	Efficiency	PA	BFA	Efficiency	PA	BFA	Efficiency
1	106.33	92.25	15.3%	1110.20	971.43	14.3%	5561.97	4882.94	13.9%
2	109.34	94.60	15.6%	1087.41	943.29	15.3%	5801.06	5056.86	14.7%
3	110.33	95.51	15.5%	1069.85	933.92	14.6%	5626.37	4887.4	15.1%
4	109.24	95.63	14.2%	1087.21	950.78	14.3%	5612.43	4876.26	15.1%
5	110.31	95.90	15.0%	1116.81	972.04	14.9%	5622.15	4893.43	14.9%
6	108.98	94.15	15.8%	1099.60	966.26	13.8%	5511.33	4799.63	14.8%
7	109.58	95.64	14.6%	1095.07	955.52	14.6%	5567.91	4846.57	14.9%
8	108.75	94.87	14.6%	1129.33	979.72	15.3%	5472.84	4811.44	13.7%
9	110.22	95.10	15.9%	1105.08	960.92	15.0%	5631.42	4918.67	14.5%
10	110.09	96.46	14.1%	1103.54	961.76	14.7%	5711.34	4988.07	14.5%
MTTT	109.32	95.01	15.1%	1100.41	959.56	14.7%	5611.88	4896.13	14.6%

**Table 3.** Simulation results for large replicas

Experiment No	600 Tasks			1200 Tasks			1800 Tasks		
	PA	BFA	Efficiency	PA	BFA	Efficiency	PA	BFA	Efficiency
1	168.58	146.59	15.0%	1647.69	1436.15	14.7%	8388.2	7309.35	14.8%
2	167.65	146.04	14.8%	1668.59	1449.71	15.1%	8309.96	7259.99	14.5%
3	160.29	140.07	14.4%	1609.33	1394.26	15.4%	8365.53	7302.63	14.6%
4	168.59	146.10	15.4%	1632.46	1427.44	14.4%	8365.92	7284.88	14.8%
5	168.32	146.34	15.0%	1659.84	1453.31	14.2%	8469.45	7399.05	14.5%
6	165.65	143.62	15.3%	1652.73	1443.91	14.5%	8423.89	7390.56	14.0%
7	159.97	139.01	15.1%	1642.65	1436.89	14.3%	8531.20	7401.64	15.3%
8	163.17	142.08	14.8%	1660.40	1452.82	14.3%	8331.64	7234.12	15.2%
9	168.73	145.65	15.8%	1658.07	1450.43	14.3%	8383.55	7229.71	16.0%
10	162.57	140.73	15.5%	1639.79	1425.14	15.1%	8464.45	7394.14	14.5%
MTTT	165.35	143.62	15.1%	1647.16	1437.01	14.6%	8403.38	7320.61	14.8%

to 3 and Figure. 2. Additionally, the efficiency of the proposed algorithm is calculated to prove its superior performance over the previous one using the following formula:

$$\text{Efficiency} = \frac{\text{PA} - \text{BFA}}{\text{PA}} \times 100 \quad (6)$$

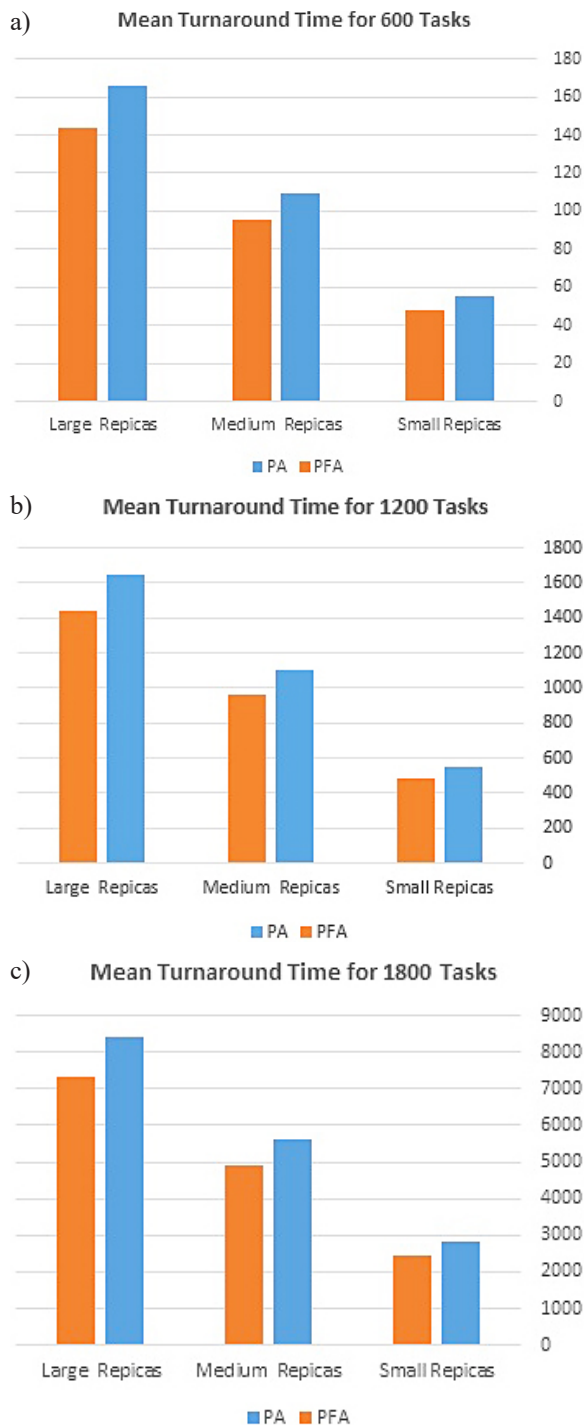
Table 1 demonstrates that in all the experiments BFA performs better than PA. The efficiency of BFA is 15.6% at its best and 14.3% in its worst case when the number of tasks is 600 while the efficiency ranges between 13.9% and 15.6% when the number of tasks is 1200 and ranges between 12.8% to 15.3 when the number of tasks 1800. The efficiencies in all experiments is slightly different which indicates the robustness of the proposed algorithm. BFA performs better in all scenarios, where the response time is reduced, the task turn-around time is reduced respectively. Furthermore,

the proposed algorithm can scale up from hundreds to thousands of tasks. Table 2 and 3 demonstrate almost similar results to Table 1, which indicates scalability in terms of replica size in addition to scalability in terms of number of tasks which proves that BFA always outperforms PA.

### CONCLUSION

The response time is the only variable that has been addressed when choosing the best replica place for the running task. Simulation results proved that the proposed algorithm surpasses the previous algorithm and can deliver the replicas to grid tasks in the shortest time. With a significant increase in performance of 12.8–16% decrease in the time needed to complete all the tasks submitted to the grid. Additionally, we found that the

best performance ( $\approx 16\%$  decrease in turnaround time) was exhibited when there was a larger number of tasks and the highest number of tasks. This in turn decreases the turnaround time for all jobs improving the whole data grid environment. The proposed algorithm can be incorporated in a real grid middleware like Globus. Our future work will focus in implementing the proposed algorithm on cloud computing.



**Fig. 2.** Replicas Sizes Verse MTTT for both algorithms; a) 600 Tasks; b) 1200 Tasks; c) 1800 Tasks

## REFERENCES

1. Jaradat A., Salleh R., Abid A., Imitating k-means to enhance data selection. *Journal of Applied Sciences*. 2009;9(19):3569–3574.
2. Ho T., Abramson D. The griddles data replication service. in *First International Conference on e-Science and Grid Computing (e-Science'05)*. IEEE, 2005.
3. Bell W.H., et al. Evaluation of an economy-based file replication strategy for a data grid. in *CCGrid 2003. 3rd IEEE/ACM International Symposium on Cluster Computing and the Grid, 2003*. Proceedings. IEEE, 2003.
4. Vazhkudai, S., Tuecke S., Foster I. Replica selection in the globus data grid. in *Proceedings First IEEE/ACM International Symposium on Cluster Computing and the Grid*. IEEE, 2001.
5. Jaradat A., et al., Accessibility algorithm based on site availability to enhance replica selection in a data grid environment. *Computer Science and Information Systems*. 2013;10(1):105–132.
6. Jaradat A., et al., Multiple users replica selection in data grids for fair user satisfaction: A hybrid approach. *Computer Standards & Interfaces*. 2020;71:103432.
7. Al-Mistarihi H.H.E., Yong C.H. On fairness, optimizing replica selection in data grids. *IEEE transactions on parallel and distributed systems*. 2008;20(8):1102–1111.
8. AL-Mistarihi H.H., Yong C.H. Response time optimization for replica selection service in data grids. *Journal of Computer Science*. 2008;4(6):487.
9. Ferdean C., Makpangou M. A scalable replica selection strategy based on flexible contracts. in *Proceedings the Third IEEE Workshop on Internet Applications, WIAPP 2003*. IEEE, 2003.
10. Sayal M., Scheuennann P., Vingralek R. Content replication in web++. in *Second IEEE International Symposium on Network Computing and Applications, 2003*. NCA 2003. IEEE, 2003.
11. Guo M., et al. 2002. A probe-based server selection protocol for differentiated service networks. in *2002 IEEE International Conference on Communications. Conference Proceedings. ICC 2002 (Cat. No. 02CH37333)*. IEEE.
12. Zhao Y., Hu Y. 2003. GRESS-a Grid Replica Selection Service. in *ISCA PDCS*. Citeseer.
13. Kim D.-H., Kang K.-W. Design and implementation of integrated information system for monitoring resources in grid computing. in *2006 10th International Conference on Computer Supported Cooperative Work in Design*. Ieee. 2006.
14. Vazhkudai S., Schopf J.M. Using regression techniques to predict large data transfers. *The International Journal of High Performance Computing Applications*. 2003;17(3):249–268.

15. Anderson T., Dahlin M. Operating Systems: Principles and Practice. Recursive books. 2014;2.
16. Park H., Lee C. Sized-based replacement-k replacement policy in data grid environments. in International Symposium on Parallel and Distributed Processing and Applications. Springer; 2006.
17. Zhou X., et al. Recon: A fast and reliable replica retrieval service for the data grid. in Sixth IEEE International Symposium on Cluster Computing and the Grid (CCGRID'06). IEEE, 2006.
18. Feng J., Humphrey M. Eliminating replica selection-using multiple replicas to accelerate data transfer on grids. in Proceedings. Tenth International Conference on Parallel and Distributed Systems, 2004. ICPADS 2004. IEEE, 2004.
19. Almuttairi R.M., et al. Intelligent replica selection strategy for data grid. in GCA 2010: proceedings of the 2010 international conference on grid computing & applications, Las Vegas, NV, 2010.
20. Almuttairi R.M., et al. Rough set clustering approach to replica selection in data grids (RSCDG). in 2010 10th International Conference on Intelligent Systems Design and Applications. IEEE, 2010.
21. Jaradat A., Amin A.H.M., Zakaria M.N. Balanced QoS replica selection strategy to enhance data grid. in International Conference on Networking and Information Technology, 2011.
22. ك.م.ر. يدي طمها, Smart Replica Selection for Data Grids using Rough Set Approximations (RSDG); 2015.
23. Grace R.K., Manimegalai R. Replica placement and predictive replica selection techniques to improve the performance of data grid. Asian Journal of Research in Social Sciences and Humanities. 2016;6(10):1824–1839.
24. Almuttairi R.M., et al., Promote Replica Management based on Data Mining Techniques. International Journal of Engineering & Technology. 2019;8(1.5):440–446.
25. Jaradat A., et al., 2012. An enhanced grid performance data replica selection scheme satisfying user preferences quality of service. European Journal of Scientific Research, 73(4):527–538.
26. Cameron D.G., et al., Analysis of scheduling and replica optimisation strategies for data grids using OptorSim. Journal of Grid Computing, 2004;2(1):57–69.
27. Sulistio A., Yeo C.S., Buyya R. A taxonomy of computer-based simulations and its mapping to parallel and distributed systems simulation tools. Software: Practice and Experience. 2004;34(7):653–673.
28. Bell W.H., et al., Optorsim: A grid simulator for studying dynamic data replication strategies. The International Journal of High Performance Computing Applications. 2003;17(4):403–416.