



Journal of Applied Sciences

ISSN 1812-5654

science
alert

ANSI*net*
an open access publisher
<http://ansinet.com>

Imitating K-Means to Enhance Data Selection

¹A. Jaradat, ¹R. Salleh and ²A. Abid

¹Faculty of Computer Science and Information Technology,
University of Malaya, 50603 Kuala Lumpur, Malaysia

²Faculty of Computer and Information Technology,
Al-Madinah International University, 40100 Shah Alam, Malaysia

Abstract: In this study, a new approach that utilizes availability, security and time as selection criteria between different replicas is proposed. However, selecting the replica in accordance with the three factors simultaneously is complicated; therefore, concepts from the K-means clustering algorithm were adopted to create a balanced (best) solution. Numerical simulations were carried out to assess the proposed technique. The results show that the proposed system outperforms random algorithm by 17% and outperforms the round robin algorithm by 11%.

Key words: K-means concepts, replica selection process, grid applications, data availability, site security

INTRODUCTION

Data grids were originally motivated by data intensive applications like projects and scientific applications that require large amounts of distributed data. In fact, the field of scientific applications allots significant effort and cost to administer and manage the huge amounts of data obtained from their simulations and experiments (Delgado and Adjoudi, 2008). Scientific applications mostly require accessing, storing, transferring, analyzing, replicating and sharing huge amounts of data in different locations around the world (Lei *et al.*, 2007).

A data grid (Foster and Kesselman, 2001) is an infrastructure which works with large amounts of data to make it possible for grid applications to share data files in a systematic way in order to provide fast, reliable, secure and transparent access to data. The aforementioned data sharing is challenging because there is a large amount of data to be shared, limited storage space, limited network bandwidth and due to the fact that the resources are heterogeneous. On the other hand, accessing a single instance of data from a single location for all users is unfeasible; it would cause an increase in data access latency. Moreover, one organization may not be able to handle such an enormous quantity of data alone. Data replication is one solution to this problem, where identical replicas of the same data are produced and stored at different distributed nodes.

Each grid site has its own capabilities and characteristics; therefore, selecting one specific site from

many sites that have the required data is an important and significant decision. This is called the replica selection decision. The replica selection problem has been investigated by many researchers who only considered response time as a criterion for the selection process. The algorithms used in the selection engine are: greedy, random, partitioned and weight algorithms (Ceryen and Kevin, 2005; Corina and Mesaac, 2003). In this study, the replica selection problem is addressed as an important decision to guarantee efficiency and to ensure the satisfaction of the grid users, providing them with their required replicas in a timely manner and in a secure way. The main contribution of this study is to produce an alternative solution to the replica selection problem encapsulated in a grid system which utilizes additional aspects in the selection process namely, availability and security. The main questions to be addressed here are: how to select the appropriate replica location from many replicas distributed across the grid sites in a short amount of time and high quality manner and how to aggregate the conflicting and heterogeneous criteria into one value for the selection. In this study, the answers to these questions are provided and an efficient grid system that outperforms the current systems in the data grid domain is demonstrated.

DATA SELECTION IN GRID ENVIRONMENT

Replication can boost the performance and strength of the grid systems and decrease data access latency by enhancing data availability and reliability. However, the

process of bringing up several copies of data introduces other problems; therefore, replicas should be selected appropriately. Replica selection is a high level service in the data grid domain that selects the best replica location. The replicas are in different grid sites where each site has its own capabilities, thus selecting an appropriate replica location is of great importance to overall system performance. Replica selection algorithms can be classified into two classes. The first class encompasses heuristically based, statically configured algorithms. The most well known algorithms in this class are: the random, round robin and proportional algorithms (Zhou *et al.*, 2006). The first two algorithms assume that all replicas have the same processing power available, while the dynamic nature of the grid environment and the response time are ignored. Another approach that falls under the same class is the proportional algorithm. In this algorithm a certain replica is selected based upon relative power ratings assigned by the system administrator; this is error prone and depends on information that has been collected in advance (offline) before the selection process takes place.

The second class of algorithms assigns clients requests dynamically according to measured conditions (i.e., selection criteria set). The famous algorithms in this class are: the greedy and weighted algorithms (Cardellini *et al.*, 2003). In the Greedy algorithm, the best replica location that offers the best criteria values is selected as a local optimum optimization and it is assumed that it leads to the global optimum. The weighted algorithm sends each request to the replica calculated to provide the best performance in accordance with the criteria (different systems adopt different metrics). Some systems begin by dividing replicas based on estimated performance, against a defined set of criteria, into two groups, available and unavailable, followed by, any selection algorithm (Corina and Mesaac, 2003).

Dynamic replica selection approaches (Ho and Abramson, 2005; Zhao and Hu, 2003) have emerged to improve the estimation of the expected user response time based on measurements of network parameters such as, network bandwidth and server request latency. An intelligent prediction based on historical log files used to decide which replica is the best. In this context, the best means the replica that has the minimum response time.

Ho and Abramson (2005), considered the dynamic changes in bandwidth, but their technique for replica selection handles only the Storage Resource Broker (SRB) datasets and does not handle other datasets such as Replica Location Service (RLS) in Globus.

Zhao and Hu (2003) have selected the GridFTP log file alone as a prediction tool to find the replica in the minimum response time, but it was clarified in (Vazhkudai and Schopf, 2002) why GridFTP alone is not enough for the best prediction. Rather a regression technique model is built for predicting the data transfer time from the source to the sink based on three data sources namely, GridFTP, Network Weather Service (NWS) and I/O Disk.

Rahman *et al.* (2005), the storage access latency is incorporated when estimating the response time relying on historical information about storage latency and data transfer times to forecast the upcoming response time. However, this forecast for storage access latency is not precise due to the grid resources changing and upgrading over time. Al Mistarihi and Yong (2008), the authors have added the jobs waiting in the queue to estimate the response time.

According to the previous literature and to the best of our knowledge, all the previous works that addressed the replica selection problem are concerned with and focused on response time only. Moreover, Munir *et al.* (2007) from the scheduling domain summarized the QoS shown in Fig. 1, where the only concern was time. However, there are factors other than time and in some cases, they are more important. Sometimes it is better to consider selecting data from secure sites rather than sites with short response times; especially if the required replicas are critical and the reality of wide area networks is that hackers, viruses and many unauthorized users attempt to violate them making them at times insecure. On the other hand, most grid sites are available daily for a limited number of hours, selecting the most available replica improves the grid system because there is more time available to resolve any problems that may arise.

Our proposed solution improves the grid environment by adding availability as a critical factor which ensures that the download will be completed from that node even if it faces difficulties like malfunctioning and degradation in bandwidth or response time. In addition to that, the solution considers the security factor which is vital for the security of the data and the whole process. The selection criteria (response time, availability

Network	Computational
<ol style="list-style-type: none"> 1. Bandwidth 2. Latency 3. Distance between nodes 4. Routing protocol 5. Queuing algorithm 	<ol style="list-style-type: none"> 1. CPU speed 2. System's memory 3. Data access and retrieval

Fig. 1: QoS parameters in a grid distributed between the network and computational aspects

and security) are heterogeneous and cannot be aggregated with each other for the selection. Furthermore, the criteria may contradict one another. Therefore, the K-means concept model was used in the selection process to solve the complexity and heterogeneity problems. Moreover, the solution uses an efficient and simple clustering algorithm (K-means) as a technique to do the complex job of selecting the best replica. The best replica is the one that shows good response time, enough availability and an acceptable level of security at the same time. Hence, the proposed selection model should consider a comprehensive view of the criteria set to make the best selection.

K-means Clustering Algorithm: Data can be divided into several non-overlapping homogenous groups that are called clusters. Each cluster contains objects that are similar amongst themselves and dissimilar to the objects of other groups. This process is called clustering and it is used in various applications in engineering statistics and numerical analysis. One of the most well known clustering tools that are used in scientific and industrial applications is the K-means algorithm. The name originates from representing K clusters S_i by mean (i.e., weighted average) S_i of its points, called the centroid (i.e., the center of the cluster). The sum of Euclidean distance $d(x,y)$ between a point (X) and its centroid (S_i) is used as an objective function as shown in Eq. 1.

$$E = \sum_{i=1}^K \sum_{x \in S_i} d^2(X, S_i) \quad (1)$$

The K-means clustering algorithm normally entails selecting a random preliminary partition or centers and repeatedly recalculating the centers based upon the partition and then re-computing the partition based on the centers. The attractiveness of the K-means algorithm is due in large part to the fact that it is simple and easy to execute. Furthermore, the K-means algorithm works with any standard norms and it is insensitive to data ordering. The K-means algorithm has some drawbacks. For example, the result strongly depends on the initial guess of centroids. In addition, the correct number of clusters is not obvious and the resulting clusters can be unbalanced. The basic K-means Algorithm by (Jain *et al.*, 1999) is outlined below:

- Choose K-data points as the preliminary centroids
- Redispense all points to their nearest centroids
- Recalculate the centroid of each newly formed cluster
- Replicate steps 2 and 3 until the centroids do not alter

When representing data with few clusters, the fine details of the data are lost, but the representation is simpler.

Approach: In this study a new centralized and decentralized data selection system (D-system) is being proposed. The new system considers the QoS of grid sites. QoS in this research is a combination of time, availability and security. This research treats time as the single best result obtained from taking into consideration all parameters shown in Fig. 1. The technique used to calculate this time is out of the scope of this study and will be future studies. The D-system can utilize many existing successful data grid core services, such as RLS and the NWS (Vazhkudai *et al.*, 2001) as shown in Fig. 2. The RLS provides the system with the physical file locations and the NWS provides information about the status of the network. The D-system selects the best site location which houses the required replica. In this context, the best site is the site that provides the highest combined security and availability and at the same time the lowest response time possible between the local site and the remote site that houses the required replica. Henceforth, we use the term “best replica” to express the highest level of QoS - for both the replica and the site which houses this replica.

The criteria set are heterogeneous and conflicting with each other, making the problem quite complex to solve. Therefore, the concepts of K-means clustering algorithm as selection factors are used to select the best replica by assuming a model site with a performance of 100% in response Time (T), 100% in Availability (A) and 100% in Security (S). This site will be named centroid. The system tries to find the closest replica to the centroid

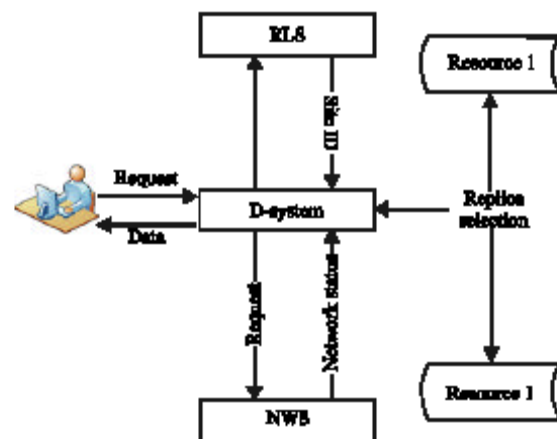


Fig. 2: Overview of the proposed system and other related entities

using the Euclidean distance. The following steps show the procedure of the (D-system):

- **Step 1:** Receives the requests from the users or typically from the Resource Broker (RB)
- **Step 2:** Gathers the replica location information from RLS
- **Step 3:** Gathers the current criteria values such as network bandwidth from the NWS
- **Step 4:** Sends a model replica $MR(T,A,S)=(100,100,100)$ with the collected replicas to K-Means Clustering Algorithm, where k is the number of replicas, to allow the model replica $MR(T,A,S) = (100,100,100)$ find the replica closest to it to form a cluster, but this approach could lead to two of the collected replicas join together and preventing the model replica from forming a cluster with any replica, especially if any two replicas are closer to each other than the model replica and its nearest replica. Hence, this step could be modified into the following two steps:

- Find the distance between the model replica $MR(T,A,S) = (100,100,100)$ and the available replicas using Eq. 2:

$$d = \sqrt{(T_1 - T_0)^2 + (A_1 - A_0)^2 + (S_1 - S_0)^2} \quad (2)$$

- Find the shortest distance between the model replica and the available replicas. Since, the distance measures heterogeneous data (t,a,s) a new unit of measurements, TAS proposed where the lowest value of TAS equals the best available site
- **Step 5:** Uses other services such as GridFTP to transport the replica with the shortest distance from the model replica to grid users securely and to make logs of the end-to-end transfer data

These steps are also represented as follows:

1. get R (file requested from the sits)
2. $i=1$
3. while R found in site
 - 3.1 get t, a, s
 - 3.2 $d(I) = \text{SQRT}((t_1 - t_0)^2 + (a_1 - a_0)^2 + (s_1 - s_0)^2)$
 - 3.3 $i = i+1$
4. best = $d(1)$
6. $j=2$
5. While $i \leq j$
 - 5.1 if $d(j) < \text{best}$
 - 5.1.1 best = $d(j)$
6. halt

RESULTS AND DISCUSSION

A user-defined number of nodes, each with varying performance in time, availability and security, were simulated to compare and verify the results of our new approach. A number of requests for certain data were carried out by the new system. The same experiments used on the new system were also performed with the same data using random algorithm and round robin algorithm. The performance of the new system showed the best results amongst the three and was the closest to the ideal model node. Thus, these findings are an added value to previous research findings, by increasing the availability and increasing the security level. Figure 3 shows the performance of the three algorithms.

Figure 3a shows the performance of the three systems with 20 requests. The results collected from the D-system consistently were found to be closer to the ideal modal value and displayed a trend to reach it. The ideal modal value is the zero line (the x-axis); the zero line indicates where the time is minimized, the availability value is maximized and the security level is maximized. The D-system selects the best node at any given time as shown in the Fig. 3, where the line represented by the D-system is almost horizontal and consistently very close

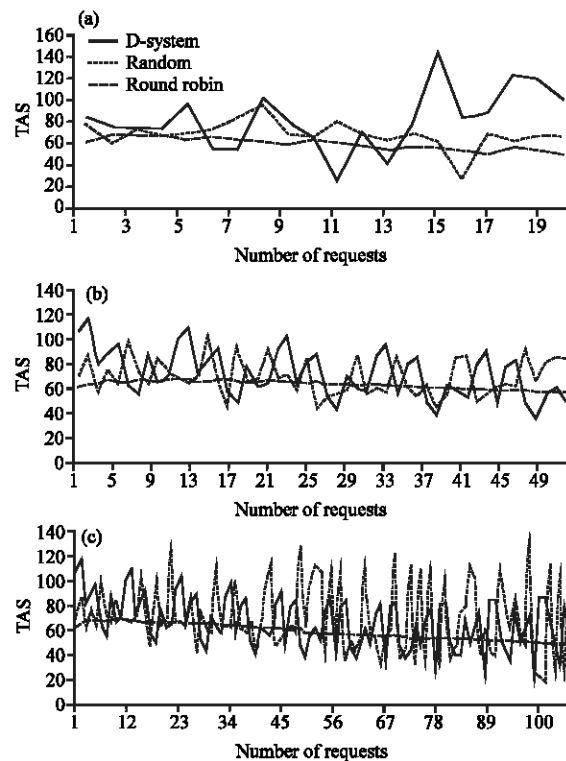


Fig. 3: (a, b, c) Performance of the three algorithms

to the x-axis. Any selected site will lose some of its power and the D-system determines the best available one for the next request. Consequently, it allows the other sites to recover or finish processing of some existing jobs and therefore release more system resources.

This ability of the D-system is an added value, in contrast with the other two systems where selection is performed by turn or random methods. If the site selected by the other two systems is overloaded or is not the best site at that particular point in time, then this will lead to a high value of TAS indicating bad performance, while allowing the best site or other better sites to release more system resources and therefore become even better. As a result, the value of TAS will become very low at some points (even lower than what we experienced from the D-system) and drastically higher at other points. This performance is clearly represented by Fig. 3a and b. The very low values of TAS presented by random and round robin algorithms unfortunately only occurred randomly or periodically, in an inefficient or unbalanced manner. In contrast, the stable and managed behavior of the D-system ensured a higher combined level of security, availability and response time.

Additionally, Fig. 3b demonstrates 50 requests during which the D-system was still functioning properly, even after increasing the number of requests, in contrast to the random behavior of the other two systems. Moreover, Fig. 3c shows 100 requests indicating that the D-system successfully scaled up to a hundred requests. The D-system even performed better as the number of requests increased. It was also noticed that the behavior of the Round Robin system showed better performance than the Random system after increasing the number of requests; this observation is most likely due to its nature of requesting from each site in a predetermined order. However, the D-system still performed better than both other systems. Figure 3 also shows that the D-system was more stable than other systems and trends to move closer to the modal ideal value (the zero line).

Furthermore, in order to demonstrate further the efficiency of the D-system over the other systems, the averages of the values resulting from the simulation were computed as shown in Table 1. The efficiency of the D-system over the other systems was computed as shown in Eq. 3:

$$\text{Efficiency} = \frac{\text{OSV} - \text{USV}}{\text{OSV}} \times 100\% \quad (3)$$

Where:

OSV = Other system value,

USV = Underlying system value

Table 1: The average values of the three systems from Fig. 3c

System	Values
D-system	57.75
Random	69.58
Round robin	64.96

Therefore, the efficiency of D-system over random equaled 17% and over round robin equaled 11%. On the other hand, a limitation of the D-system occurs if there are two or more sites which have the best possible combination of capabilities and levels of performance in terms of time, security and availability, which are equal in proportional value but vary in the order, then the D-system will randomly select one of them by default. This is due to the fact that no constraints exist in the D-system to give preference to one factor over another. An example of this behavior can be seen in the following data: Site A has the following values: Security = 90, Availability = 80 and Time = 75, while site B has the following values: Security = 75, Availability = 90 and Time = 80. However, resolving this issue is beyond the scope of this study. The D-system takes new values (i.e., security and availability) into consideration during the selection process making it more efficient, secure and reliable than the previous systems.

The D-system ensures that it will be more secure than previous systems (where security was ignored) and at the same time, it is more consistent and reliable since it will consider the availability of the selected site. The cost of these advantages is the possible negative impact of response time, which previous systems focus on.

CONCLUSION

In conclusion, the proposed system, namely the D-system, performed better than both the Random and Round Robin Algorithms. New factors, namely security and availability, were considered in the D-system. Therefore, the distance in the D-system in replica selection was very close to zero, as shown in the results, indicating enhanced system performance. The D-system can be of much benefit to other grid services that require data selection in data grid environments, utilizing less time and demonstrating high quality performance.

REFERENCES

- Al Mistarihi, H.H.E. and C.H. Yong, 2008. Replica management in data grid. *IJCSNS*, 8: 22-32.
- Cardellini, V., M. Colajanni and P.S. Yu, 2003. Request redirection algorithms for distributed web systems. *TPDS*, 14: 355-368.

- Ceryen, T. and M. Kevin, 2005. Performance characterization of decentralized algorithms for replica selection in distributed object systems. Proceedings of 5th International Workshop on Software and Performance, July 11-14, Palma, de Mallorca, Spain, pp: 257-262.
- Corina, F. and M. Mesaac, 2003. A scalable replica selection strategy based on flexible contracts. Proceedings of the 3rd IEEE Workshop on Internet Applications, June 23-24, IEEE Computer Society Washington, DC, USA., pp: 95-99.
- Delgado, J. and M. Adjouadi, 2008. Towards an efficient and extensible grid-based data storage solution. Proceedings of 22nd International Conference on Advanced Information Networking and Applications, March 25-28, Florida Int. Univ. Miami, pp: 659-666.
- Foster, I. and C. Kesselman, 2001. The anatomy of the grid: Enabling scalable virtual organizations. *Int. J. High Perform. Comput. Appl.*, 15: 200-222.
- Ho, T. and D. Abramson, 2005. The griddles data replication service. Proceedings of the 1st International Conference on E-Science and Grid Computing, Dec. 5-8, Melbourne, Australia, pp: 271-278.
- Jain, A.K., M.N. Murty and P.J. Flynn, 1999. Data clustering: A review. *ACM Comput. Surveys*, 31: 264-323.
- Lei, M., S.V. Vrbsky and Q. Zijie, 2007. Online grid replication optimizers to improve system reliability. Proceedings of IEEE International, IPDPS 2007, March 26-30, Dept. of Comput. Sci., Alabama Univ., pp: 1-8.
- Munir, E.U., J. Li and S. Shi, 2007. QoS sufferage heuristic for independent task scheduling in grid. *Inform. Technol. J.*, 6: 1166-1170.
- Rahman, R.M., K. Barker and R. Alhaji, 2005. Replica selection in grid environment: A data-mining approach. Proceedings of the 2005 ACM Symposium on Applied Computing, March 13-17, Santa Fe, New Mexico, pp: 695-700.
- Vazhkudai, S., S. Tuecke and I. Foster, 2001. Replica selection in the globus data grid. Proceeding of the First IEEE/ACM International Symposium, May 15-18, Brisbane, Australia, pp: 106-113.
- Vazhkudai, S. and J.M. Schopf, 2002. Using regression techniques to predict large data transfers. *IJHPCA*, 17: 249-268.
- Zhao, Y. and Y. Hu, 2003. GRESS a grid replica selection service. Proceeding of the 16th International Conference on Parallel and Distributed Computing Systems, Nov. 3-5, USA, ACTA Press, pp: 423-429.
- Zhou, X., E. Kim, J.W. Kim and H.Y. Yeom, 2006. ReCon: A fast and reliable replica retrieval service for the data grid. Proceedings of the 6th IEEE International Symposium on Cluster Computing and the Grid, May 16-19, Singapore, pp: 446-453.