New Replica Selection Technique for Binding Replica Sites in Data Grids

Rafah M. Almuttairi, Rajeev Wankar,
Atul Negi, C. R. Rao.
Department of Computer and Information Sciences
University of Hyderabad
Hyderabad,500046, A.P. India.
rafahmohammed@gmail.com
{wankarcs, atulcs, crrsm}@uohyd.ernet.in

Mahdi S. Almahna
Department of Computer science &Engineering
Acharya Nagarjuna University
Nagarjuna Nagar-522524
A.P.India
mehdisaleh@gmail.com

***Abstract*—The objective in Data Grids is to reduce access and file (replica) transfer latencies, as well as to avoid single site congestion by the numerous requesters. To facilitate access and transfer of the data, the files of the Data Grid are distributed across the multiple sites. The effectiveness of a replica selection strategy in data grids depends on its ability to serve the requirement posed by the users' jobs. Most jobs are required to be executed at a specific execution time. To achieve the QoS perceived by the users, response time metrics should take into account a replica selection strategy. Total execution time needs to factor latencies due to network transfer rates and latencies due to search and location. Network resources affect the speed of moving the required data and searching methods can reduce scope for replica selection. This paper presents a replica selection strategy that adapts its criteria dynamically so as to best approximate application providers' and clients' requirements. We introduce a new selection technique (EST) that shows improved performance over the more common algorithms.**

Keywords: Data Grid; Replica Selection Technique; Hungarian algorithm; association rules.

I INTRODUCTION

Grid Computing emerges from the need to integrate collection of distributed computing resources to offer performance unattainable by any single machine [15]. *Data Grid* technology facilitates data sharing across many organizations in different geographical locations as it is shown in *Figure 1*. *Data Replication* is a service to move and cache data close to users. It is a solution for many grid-based applications such as climatic data analysis and physics grid network [6] which both require responsive navigation and manipulation of large-scale datasets. Moreover, if multiple replicas exist, *Replica Management Service (RMS)* is required to discover available replicas and select the best replica that matches the user's requirements. To collect all logical names of replicas and their locations *Replica Location Service (RLS)* is used. To serve the user's request with best one, *replica selection strategy* is used [17].

Since there is more than one replica of the requested file at the run job time, the best replica selection becomes an important decision because it affects the efficiency of execution [8]. Previous selection strategies like random and round robin have limitations as their selection does not depend upon the characteristics of replicas or their network links status [18].

To cover those limitations, here we propose a new replica selection technique that uses association rules of data mining approach. We use *Apriori* algorithm for this purpose which combines different associated sites, having uncongested links at the time of the file(s) transfer.

The rest of the paper is organized as follows: *Section II* summarizes the related work. *Section III* contains preliminary concepts of association rules. *Section IV* explains the general aspect of the data grid architecture. Our proposed technique is explained in *Section V*. Simulation input is shown in *Section VI* and the results and their interpretation are presented in *Section VII* then we conclude in *Section VIII*.

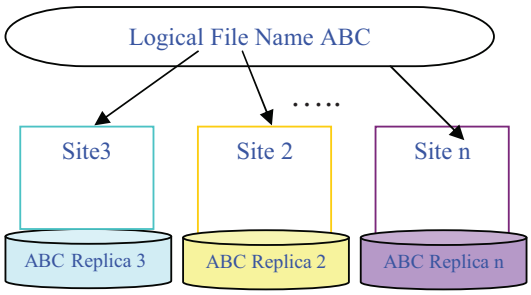

Figure1. Replicas of ABC file

II RELATED WORK

Replica Selection Problem (RSP) has been investigated by many researchers who only considered response time as a criterion for the selection process. *F. Corina* and *M. Mesaac* [24] in 2003 *and Ceryen* and *M. Kevin* [23] in 2005 used different algorithms such as greedy, random, partitioned and weight algorithms in the selection engine.
The first replica selection approaches proposed to bind a client to the nearest replica, with respect to some static

metric such as the geographical distance in miles [2, 9] and the topological distance in number of hops [26]. However, as several experimental results [27,28] show, the static metrics are not good predictors for the expected response time of client requests. The main drawback of both geographical and topological network metrics is that they ignore the network path's dynamic conditions.

In 2001, *R. Kavitha,* and *I. Foster*. [3], used traditional replica catalog based model, where for each new request Replica Location Service is queried to get the addresses of replica's sites and then probe the network link using Hop count method to select the best replica. This way of selection is not efficient because the number of hops does not reflect the actual network condition such as Network Bandwidth and link's latency.

During 2001-2003, *Sudharshan* et al. [4,5,8] contributed many research results. In their work they used the history of previous file transfer information to predict the best site holding a copy of the requested file. When a file transfer has been made between two sites, the file size, the available network bandwidth, and transfer time are saved, thus it can be used later for training and testing the regression model to predict the actual transfer time. In their work they showed that data from various sources can help in better predictions than data from one source. They achieve a better accuracy in file transfer throughput prediction by using data from all of these three sources: data streams of network, file size, and past grid transfer information.

In 2005, *Rashedur* et al. [1] exploited a replica selection technique with the *K-Nearest Neighbor (KNN)* rule used to select the best replica from the information gathered locally. The *KNN* rule selects the best replica for a file by considering previous file transfer logs indicating the history of the file and those similar. This technique has a drawback as they mentioned in their paper: the misclassification will increase in case of the large file transfer and will cost more than a couple of small file transfer misclassifications. Especially in the Gaussian random access pattern the accuracy is the lowest. Another drawback in *KNN* is that one needs to save all previous instances (file requests) to use them to select the best replica site, which means it will take some time to search in the large history of data base and the result might or might not be correct.

In 2008, *Rashedur* et al. [17] proposed a *Neural Network* predictive technique (NN based) to estimate the transfer time between sites. The predicted transfer time can be used as an estimate to select the best replica site among different sites. Simulation results demonstrate that *Neural Network* predictive technique works more accurately than the multi-regression model, which was used before *NN* [4,8,5]. However *NN* technique does not always give the right decision because the copy of the file may become no longer available (this is a common occurrence in grid) in the predicted site, so in this case the *Traditional Model* has to be used.
In 2009, *A. Jaradat* et al. [18] proposed a new approach that utilizes availability, security and time as selection criteria between different replicas, by adopting k-means clustering algorithm concepts to create a balanced (best) solution. The best site does not mean the site with shortest time of file transfer, but the site which has three accepted values: security level, availability and time of file transfer.

In our previous work we first proposed the association rule mining approach to RSP [25]. Here, in this paper we compare our strategy with random selection strategy. We are introducing a usage of the other simulation software called *XLMiner* [20] to get association rules using Apriori Algorithm. This study improves the replica selection decision to achieve higher efficiency and to ensure the satisfaction of the grid users, providing them with their required replicas in a timely manner.

III PRELIMINARY CONCEPTS

In this section we declare the preliminary concepts of *Data Mining DM*:

A) Association rules: association rules are mostly used in mining transaction data. Crucial terms in association rules terminology are:
- *Item* (in *DM* terminology corresponds to attribute-value pair)
- *Transaction* (a set of items; corresponds to example)
- *A Set* (data set) of transactions containing more different items

For the transactions it is typical that they differ in the number of items. Therefore, some transformations (standard form) as it is shown in *(Table I)* might be necessary to be able to data mine transaction data with one of the data mining tools [21].

Each transaction in the set gives us information about which items co-occur in the transaction. Using this data one can create a co-occurrence table that tells the number of times that any pair (or itemset) occurs together in the set of transactions. From the co-occurrence table we can easily establish simple rules like:

Rule 1= "*Item 1* comes together with *Item 2* in *10%* of all transactions"

In this rule, the *10%* is a measure of the number of co-occurrences of these two items in the set of transactions, and is called a *support* of the rule. If the frequency of *Item 1* occurring in the set of transactions is *10%*, and that of *Item 2*, *20%*, then the ratio of the number of transactions that support the rule (10%) to the number of transactions that support the *Antecedent* part of the rule gives the *confidence* of the rule. In this case the *confidence* is:

Rule 1= "*Item 2* comes together with *Item 1* in *10%* of all transactions"

Confidence of this rule is:

c(Rule 1)=10/20=0.5

So, the confidence of the *rule 1* is *0.5* and is equivalent to saying that when *Item 2* occurs in the transaction, there is a *50%* chance that also *Item 1* will occur in the transaction. The most confident rules seem to be the best ones. But the problem arises, for example, if *Item 2* occurs more frequently in the transactions (let's say in *60%* of transactions). In that case the rule might have lower confidence than the random guess! This suggests using another measure called *improvement.* That measure tells how much better a rule is at predicting the *Consequent* than just assuming the result. *Improvement* is given by formula [22]:

$$I(Rule\#) = \frac{p(Antecedent \wedge Consequent)}{p(Antecedent) * p(Consequent)} \quad (1)$$

*I(Rule 1)=0.1/(0.1*0.2) = 5.*

When improvement is greater than 1 the rule is better than the random chance. When it is less than 1, it is worse. In our case *Rule 1* is five times better than the random guess.

IV DATA GRID ARCHITECTURE

In this section a Data Grid architecture *PRAGMA* (see *Figure 2*) is explained with functionality of each component.

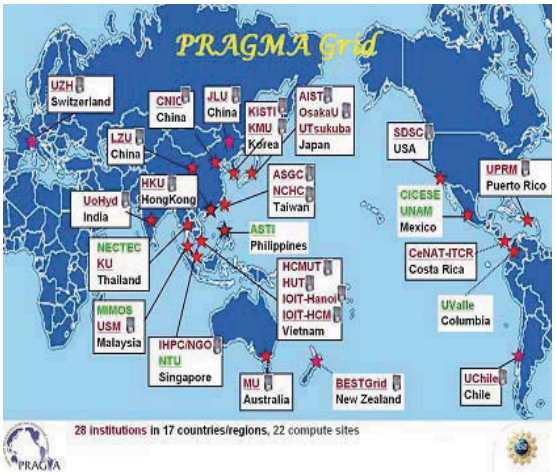

Figure 2. PRAGMA Grid, 28 institutions in 17 regions

A) Replica Management System (RMS)

As we see in the *Figure 3*, the main component of the Data Grid is the *Replica Management System (RMS)* it acts as a logical single entry point to the system and interacts with the other components of the systems as follows:

B) Replica Location Service (RLS)

Replica Location Service (RLS) is the service that keeps track of where replicas exist on physical storage systems. It is responsible for maintaining a catalog of files registered by the users or services when the files are created. Later, users or services query *RLS* servers to find these replicas.

Before explaining *RLS* in details, we need to define a few terms, such as:

- A *Logical file Name (LN)* is a unique identifier for the contents of a file.
- A *Physical file Name (PN)* is the location of a copy of the file on a storage system.

These terms are illustrated in *Figure 1*. The job of *RLS* is to maintain associations or mappings between logical file names and one or more physical file names of replicas. A user can provide a logical file name to an *RLS* server and ask for all the registered physical file names of replicas. The user can also query an *RLS* server to find the logical file name associated with a particular physical file location. In addition, *RLS* allows users to associate attributes or descriptive information (such as size or checksum) with logical or physical file names that are registered in the catalog. Users can also query *RLS* based on these attributes [8].

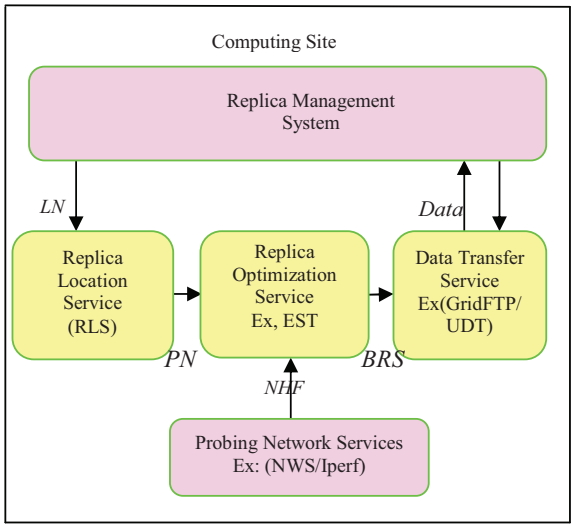

Figure3. Functionality of Replica Management System

C) Replica Optimization Service (ROS)

The Optimization component is used to minimize file access times by pointing access requests to appropriate replicas and replicating frequently used files based on gathered access statistics. The goal of the optimization service is to select the *Best Replica Site (BRS)* with respect to a network and storage access latencies [19]. *ROS* gathers the information from the network monitoring service like *Network Weather Service (NWS)* [11]; or *Iperf Service* [10] and the storage element service about the respective data access latencies.

D) Data Transfer Service (DTS)

After physical addresses are known, *RMS* asks *DTS* to transfer the requested file sets using a high- performance, secure and reliable data transfer protocol like *GridFTP* [13] and *UDT* [12]. After getting a simple and clear picture about the infrastructure of the data grid, next section explains

where our model resides and how much it changes the data grid performance.

V EFFICIENT REPLICA SET SELECTION TECHNIQUE

This section clarifies our approach, its performance, its difference from the other models and what are the helpful advantages for us to cover the limitations of the traditional and random selection methods. Let us explain the main steps of our strategy.

- *Single Trip Time STT.*
  STT is a time taken by the small packet to travel from Replica's Site (RS) to Computing Site (CS). The *STT* delays include packet-transmission delays (the transmission rate out of each router and out of the replica site), packet-propagation delays (the propagation on each link), packet-queuing delays in intermediate routers and switches, and packet-processing delays (the processing delay at each router and at the replica site) for a single trip starting from replica site to the computing site. It means that *STT* is the summation of all these delays. We use Standard deviation of *STT* as a factor to check the stability or instability of the network links [14]. Before selection process starts, the computing site receives periodical *STT*s of all replicas' sites and stores the most recent in a log file called *Network History File (NHF)* as it is shown in *Table I.*
- *Standardization Data.*
  Using a mapping function we can convert *STT* values to logical values and safe the result in *Logical History File (LHF)* as it is shown in *Table II.*
- *Association Rules Discovery.*
  One of popular association rules algorithms of data mining approach is *an Apriori algorithm.* Here, it is used for discovering associated replica sites to work concurrently and minimize total time of transferring the requested file(s) as it is shown in *Figure 4* [16].
- *Evaluation Rules.*
  Evaluation process is needed to check the validity of the association rules.
- *Efficient Set algorithm.*

In this section, we declare the steps of our proposed algorithm to get the best set of replica sites working concurrently with the minimum cost of getting the requested files.

Step I Receive a job from User/Application.
Step II Contact *RLS* to get all replica names.
Step III Contact *NWS/ Ipref* to get a *NHF.*
- Rows = STTs
- Columns = Replica Sites

Step IV Convert *Network History F*ile *(NHF)* to *Logical History File (LHF)* that contains logical values (*LV*) applying the following mapping function for each column :

Step V Convert *Network History F*ile *(NHF)* to *Logical History File (LHF)* that contains logical values (*LV*) applying the following mapping function for each column :

a) Calculate the Mean:

$$MSTT_{i,j} = \frac{(\sum_{k=i}^{(l-1)+i} STT_{k,j})}{l}, \quad where \quad l = 10$$

b) Calculate the Standard deviation:

$$STDEVi,j = \sqrt{\frac{\sum_{K=i}^{(l-1)+i} (STT_j - MSTT_{i,j})^2}{l}}$$

c) *Find* $Q_{i,j} = \frac{STDEV_{i,j}}{MSTT_{i,j}} * 100$

d) *Find* $AVi = \frac{\sum_{j=1}^{M} Q_{i,j}}{M}, \quad where M = number replicas$

e) *Compare* $IF(AVi < Q_{i,j}) \quad then LV = 0 \, otherwise LV = 1$

Step VI Apply an *Association Rules Technique* (*AT*), such as *Apriori* algorithm[21]
Call *AT (LHF,c,s,AR)*
Input:
LHF: Logical values of Network History File
c: Minimum confidence value.
s: Minimum support value.
Output:
AR: Association Rules

Step VII Measure rule's correlation using an Improvement equation:

$$I(Rule\#) = \frac{p(Antecedent \wedge Consequent)}{p(Antecedent) * p(Consequent)}$$

If (I <1) this indicates negative correlation
Otherwise it is positive correlation.

Step VIII Send physical names of the highest correlation rule sites to the transport service such as (*GridFTP/UDT*) in order to get the requested files.

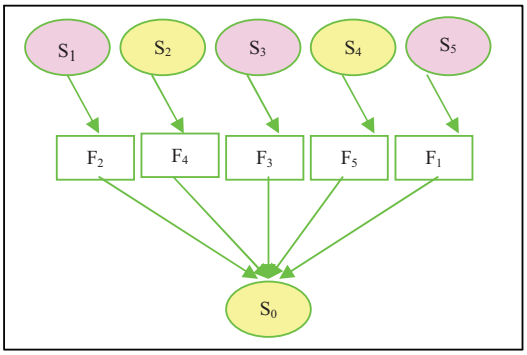

Figure 4. Multiple sites concurrently send different files

VI SIMULATION INPUTS:

In our simulation, we suppose that all replicas have the same characteristics, such as number of jobs to be run, file processing time, delays between each job submission, maximum queue size in each computing element, size and number of requested files and speed of input/output storage operations to see the effect of network resources only. We tested and compared the two selection strategies with our strategy, traditional and random models.

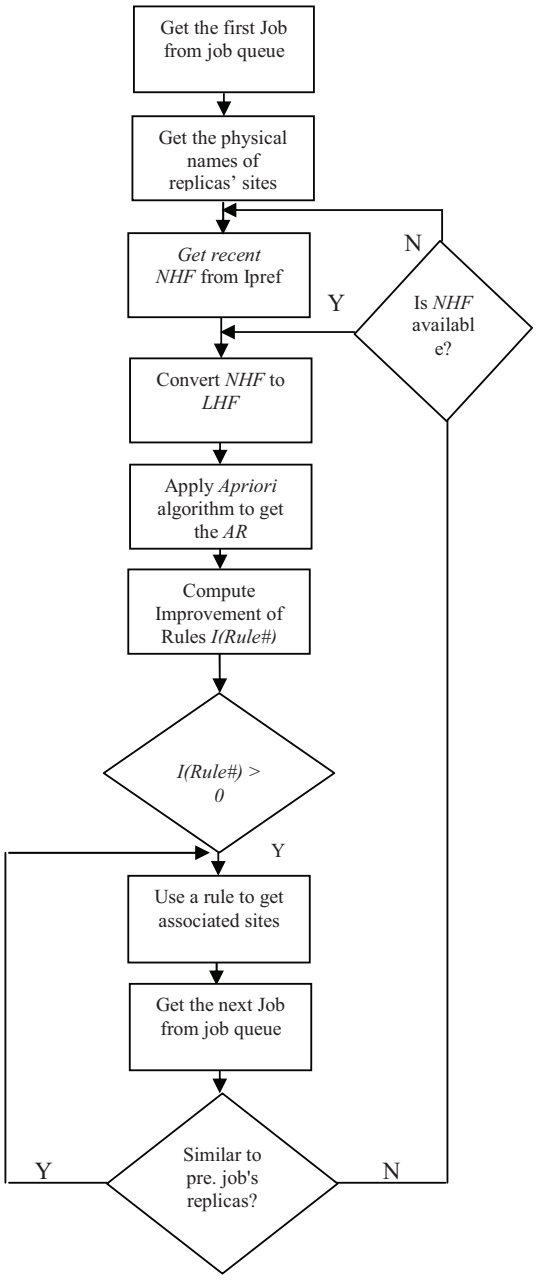

Figure 5. Flowchart of ESM.

In traditional selection method, the best replica is the one which has the least number of hops (routers), the highest bandwidth or the minimum round trip time to reach the computing site. In random method replica manager selects randomly one of the available replicas to serve user's request. Figure 4, shows the flowchart of *EST* steps.

A) Get data grid job: in our simulation we assume that there is a job (*J*) with four files required for analyzing.

B) By contacting *RLS* all Logical and Pysical names of replicas are collected.

C) *Getting Single Trip Time STT.* Being a node of *PRAGMA* Data Grid [7], we construct *NHF* using *Ipref* service of *PRAGMA* infrastructure. We contact data grid nodes and get the *STTs* for different periods of time and save *STTs* values in a file called *NHF.xls* as it is shown in *Table I.* Therefore, our grid configuration file reflects the real network nodes and links of the *PRAGMA Grid* [7].

D) Convert *STTs* values to logical values using a mapping function.

E) *Applying Association Rules.* To apply an *Apriori* algorithm on logical values of *STT Table*, *Table II* we use *MLMiner* software [20]. The steps which we followed are:

1. Use a spreadsheet software to open *NHF.xls* file.
2. From the *XLMiner* menu, select *Add-Ins* menu then *Affinity* after that chose *Association rules*. The Association Rules dialog box appears as it is shown in *Figure 6* then selects the input data format as "Data in binary matrix format".
3. Enter all input data in the *Association Rules* box such as minimum confidence and support and make the selections of the binary data of *STT* and click Finish button. The result will be shown as in *Figure 7.*

TABLE I. TRANSACTIONS TABLE, STTS VALUES

STT	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8
1	60	176	202	336	78.7	256	76.4	213
2	65	211	209	343	76.5	256	86.8	202
3	303	175	298	338	64.8	258	57.2	205
4	305	213	203	273	92.6	255	85.1	202
5	300	210	223	334	95	292	55	212
6	313	176	207	335	66.7	298	55.3	212
7	310	175	298	273	94.2	255	85.8	202
8	307	216	260	271	94.4	256	60	212
9	310	217	260	342	95.2	289	90.1	212
10	310	211	224	339	66.3	257	90.3	212
11	310	175	204	272	92.8	262	91.4	202
12	307	176	205	271	69	256	88.7	210
13	310	211	227	344	92.5	299	64.3	212
14	50	175	202	270	66.3	299	90.5	216
15	316	214	260	336	63.4	296	57.8	224
16	74	209	206	341	94.3	287	56.4	222

TABLE II. LOGICAL VALUES, STANDARD FORM

STT	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8
1	0	1	0	1	0	1	0	1
2	0	1	0	1	0	1	0	1
3	1	1	0	0	0	1	0	1
4	1	1	0	0	0	1	0	1
5	0	1	1	1	0	1	0	1
6	0	1	1	1	0	1	0	1
7	0	1	1	1	0	1	0	1
8	0	1	1	1	0	1	0	1
9	0	1	1	1	0	1	0	1
10	0	1	1	1	1	1	0	1
11	0	1	1	1	1	1	0	1
12	0	1	1	1	1	1	0	1
13	0	1	1	1	1	1	0	1
14	0	1	1	1	1	1	0	1
15	0	1	1	1	1	1	0	1
16	0	1	1	1	1	1	0	1

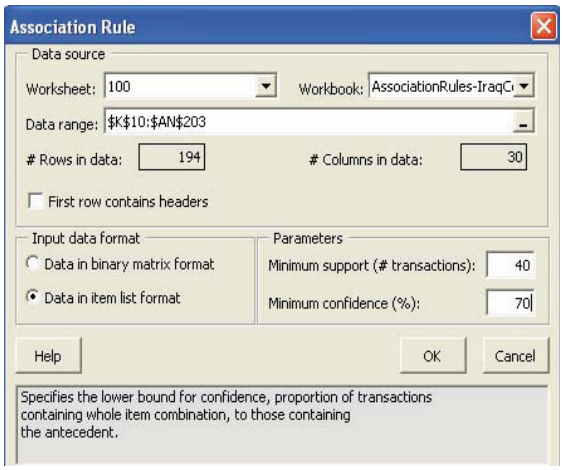

Figure 6. Association Rules window in XLMiner software

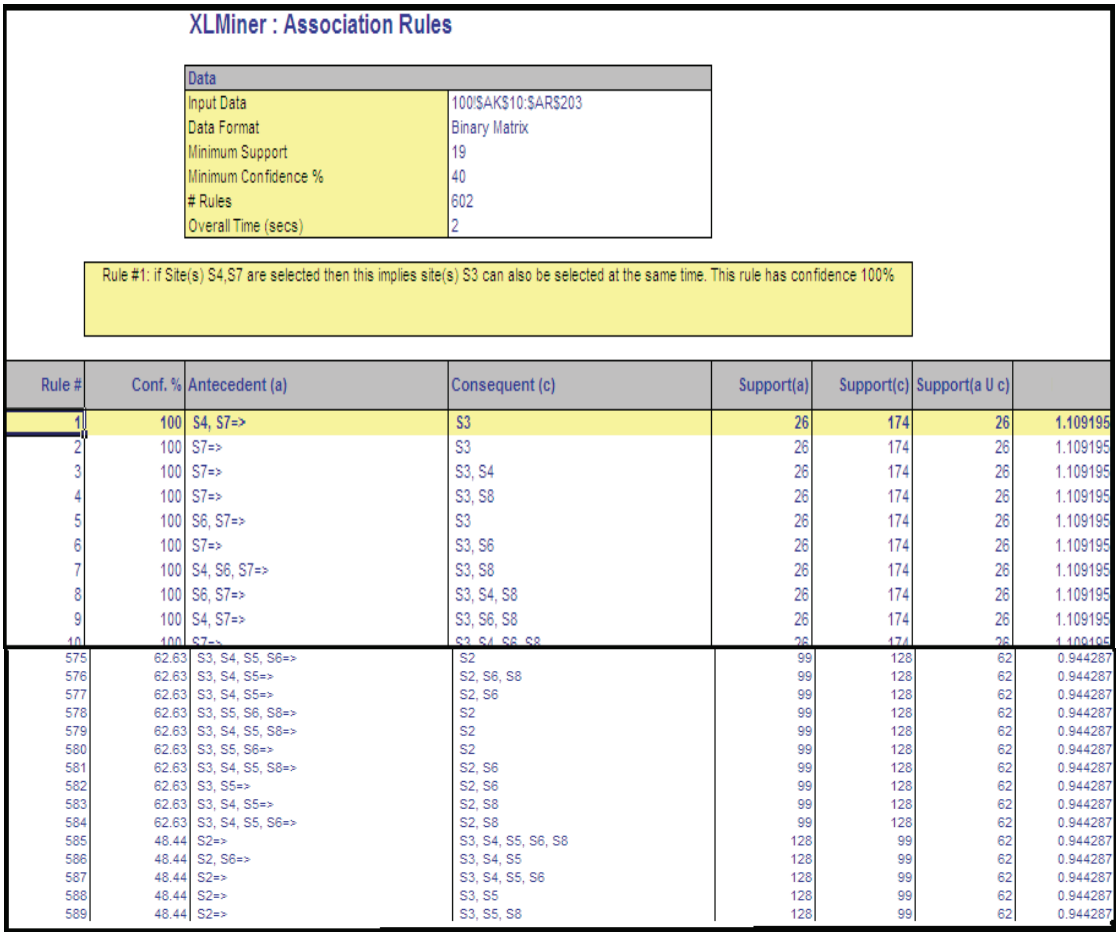

Figure 7. XLMiner menue of Association Rules

F) To check the validity of association rules *Equation 1* is used as it is done in the last column of *Figure 7*.

G) Select one of the rules which have Improvement value more than (*1*).

H) In case if there is another job asking to get the files, and these files are available in the same sites then choose another rule to serve the new request. Otherwise apply *Apriori* algorithm for recent STTs of new replicas sites.

VII INTERPRETING THE RESULTS

This section means to explain how the association rules work better than the traditional and random methods. As it is shown in *Figure 7*, after applying *Apriori* algorithm we get *602* different rules which can be used to select the best combination of replica sites. Let us explain Figure 7 in details.

"The rule": *Rule #1: if Site(s) S_4, S_7 are selected then this implies that site(s) S_3 can also be selected at the same time. This rule has 100% confidence.*

In other words, it means if site S_4, and S_7 are selected to work together to transfer the requested files, then this implies site(s) S_3 can also be selected to share the work at the same time. This rule has confidence *100%*. This particular rule has *confidence* of *100%*, meaning that, S_4, S_7, and S_3 can be selected as a best set of replicas by *Replica Manager* to get requested files. To compute the correlation of this rule and see how far it is better than choosing the site randomly, we use an Improvement equation:

"**Support (a)**" indicates that it has support of *26* transactions, meaning that in transaction *Single Trip Time Table* there are *26* concurrent uncongested trips of (S_4, S_7) i.e. these sites have similar network conditions in particular time.

"**Support (c)**" indicates the total number of transactions involving uncongested trips of S_3, in *Rule 1* is equal to 174. (This is a piece of a side information; it is not involved in calculating the confidence or support for the rule itself.)

"**Support (a U c)**" is the number of transactions where *(S_4, S_7)* as well as *(S_3)* has uncongested trips. In *Rule 1* it is equal to *26*.

"**Improvement ratio** or **Lift ratio**" indicates how much more likely we are to encounter S_4 and S_6 transaction if we consider just those transactions where S_3, S_5, and S_8 have uncongested trips. As compared to the entire population of the transactions, it's the ***confidence*** divided by ***support (c)*** where the latter is expressed as a percentage.

For *Rule 1*, the ***confidence*** is *100%* ***support (c)*** (in percentage) = *(174/194)*100 = 89.69*. So, the *Lift ratio = 100/89.69.1 = 1.1.*

As it is clearly shown in *Figure 7* some rules with an improvement value less than one means this is an unreliable rule. Whereas the rule with a value more than one means this rule is better than random replica selection with number of time equal to improvement value as it is shown in *Figure 8*.

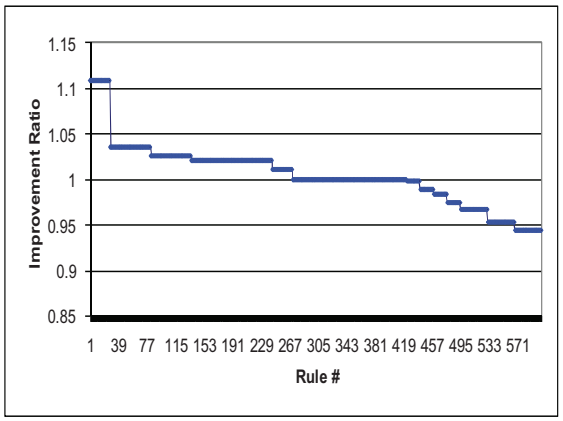

Figure 8. Improvement ratio for different rules

When improvement value is more than 1 it is better to use EST to select replica sites, because it selects the sites able to work simultaneously.

In *Figure 9* we show the comparison between EST and traditional model using highest bandwidth as a criterion to select the best replica. As we can observe our technique has a better performance most of the times because it selects the sites which have the stable links. In traditional method the site which has the highest bandwidth does not always mean to be the best because sometimes this highest bandwidth link can be congested. Let us declare more by the following scenario of *Figure 10*, suppose *(S_0)* be the computing site and let *{S_1, S_3, S_{14}}* be replica sites. Red stars referring to congested routers. Using traditional selection method the file will be got from S14 since it has less number of Hops (routers) and highest and also has highest bandwidth link.

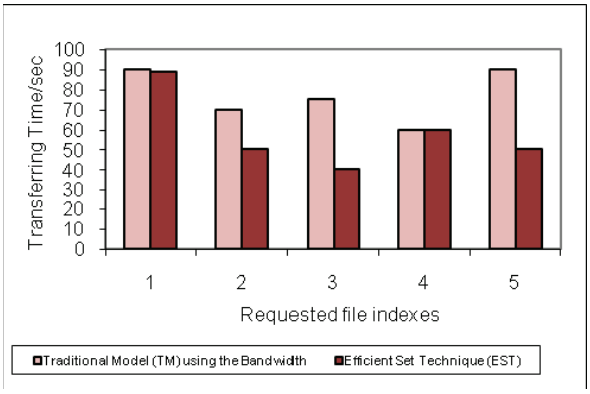

Figure 9. Traditional selection strategy and EST

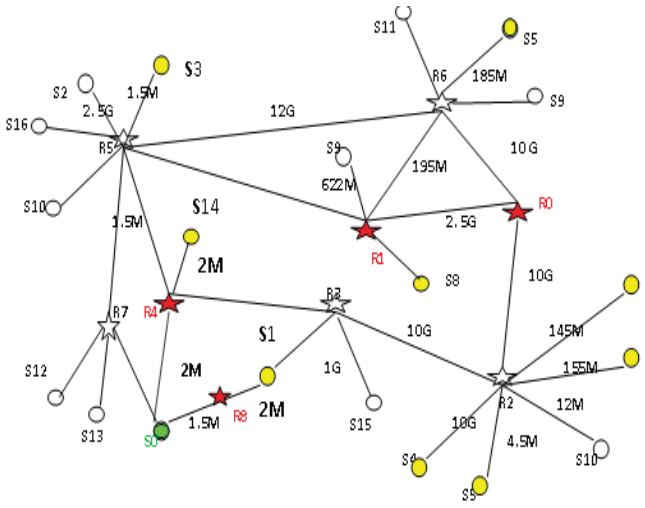

Figure 10 Data Grid and their associated network geometry

Using *EST*, the replica (S_3) is selected as a best replica because the link between *CS* and *RS* is uncongested .

VIII CONCLUSION

In this paper we presented a dynamic replica selection strategy that aims to adapt at run-time its criteria to flexible QoS binding contracts specified by the service provider and/or the client. The adaptability feature addressed by our replica selection strategy is inferred from the observation that the basic metrics, which influence the QoS that the user perceives when accessing a replica, depend directly on the application being replicated and on the clients' preferences. To reach this objective that, we used the concept of association rules of data mining approach to the most stable links sites in order to reduce the searching space the response time and network resources consumed.

ACKNOWLEDGEMENTS

Authors wish to express their sincere thanks to Prof. Arun Agarwal, from GridLabs Department of Computer and Information Sciences, University of Hyderabad, India for providing all the infrastructural and computational support required to carry out this work. His academic suggestions to improve the quality of the work are also highly appreciated and acknowledged.

REFERENCES

[1] M. Rashedur Rahman, Ken Barker, Reda Alhajj, "Replica selection in grid environment:a data-mining approach", Distributed systems and grid computing (DSGC),pp: 695 – 700 , 2005

[2] J. Gwertzman and M. Seltzer, "The case for geographical push cashing", In Proceeding of the 5th Workshop on Hot ZTopic in Operating Systems, 1995.

[3] R. Kavitha, I. Foster, "Design and evaluation of replication strategies for a high performance data grid", in, Proceedings of Computing and High Energy and, Nuclear Physics, 2001.

[4] S. Vazhkudai, J. Schopf, I. Foster, "Predicting the performance of wide-area data transfers", in: 16th International PDPS, 2002.

[5] S. Vazhkudai, J. Schopf, "Using regression techniques to predict large data transfers", in: Computing: Infrastructure and Applications, The International Journal of High Performance Computing Applications, IJHPCA , August, 2003.

[6] A. Abbas, Grid Computing:" A Practical Guide to Technology and APPLICATIONS", 2006.

[7] http://goc.pragma-grid.net/wiki/index.php/UoHyd.

[8] S. Vazhkudai, S Tuecke, I. Foster, "Replica selection in the globus data grid", in: First IEEE/ACM International Conference on Cluster Computing and the Grid, CCGrid 2001.

[9] J. Guyton and M. Schwartz."Locating nearby copies of replicated internet servers", In Proceeding of ACM SIGCOMM' 95, 1995.

[10] A. Tirumala, J. Ferguson, Iperf 1.2 - The TCP/UDP Bandwidth Measurement Tool, 2002.

[11] R. Wolski, Dynamically forecasting network performance using the Network Weather Service, Cluster Computing (1998).

[12] Yunhong Gu, Robert L. Grossman, "UDT: UDP-based data transfer for high-speed wide area networks", Computer Networks, Volume 51, Issue 7, 16 May 2007, Pages 1777 1799. Elsevier.

[13] R.M. Rahman, K. Barker, R. Alhajj, "Predicting the performance of GridFTP transfers", in: Proceedings of IEEE Symposium of Parallel and Distributed Systems, 2004, New Mexico, USA, p. 238a.

[14] J. F. Kurose, K.W. Ross, "Computer Networking A Top-Down Approach Featuring the Internet", 3rd edition.

[15] S. Venugopal, . R. Buyya,"The Gridbus Toolkit for Service Oriented Grid and Utility Computing: An Overview and Status Report"2004.

[16] R. Agrawal, T. Imielinski, A.Swami, "Mining association rules between sets of items in large databases". In: Proc. ACM SIGMOD Intl. Conf. Management Data, 1993

[17] R.M Rahman, K Barker and R Alhajj, "Replica selection strategies in data grid", Journal of Parallel and Distributed Computing, Volume 68, Issue 12, Pages 1561-1574, December 2008.

[18] A. Jaradat, R. Salleh and A. Abid, "Imitating K-Means to Enhance Data Selection", Journal of Applied Sciences 9 (19): 3569-3574, 2009, ISSN 1812-5654, Asian Network for Scientific Information-2009.

[19] S. Venugopal, . R. Buyya, K. Ramamohanarao, "A taxonomy of Data Grids for distributed data sharing, management, and processing". ACM Comput. Surv. 38, 1 (Jun. 2006), ACM New York, NY, USA

[20] http://www.resample.com/xlminer/help/Index.htm

[21] A. K Pujari, "Data mining techniques", Hyderabad : Universities Press, 2002.

[22] G. Williams, M. Hegland and S. Roberts, "A Data Mining Tutorial", IASTED International Conference on Parallel and Distributed Computing and Networks (PDCN'98) 14 December 1998.

[23] T. Ceryen, and M. Kevin, 2005. "Performance characterization of decentralized algorithms for replica selection in dstributed object systems". Proceedngs of 5th International Workshop on Software and Performance, July 11 -14, Palma, de Mallorca, Spain, pp: 257-262.

[24] F. Corina, 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.

[25] R. M. almuttari, R. Wankar, A. Negi, C.R. Rao. "Intelligent Replica Selection Strategy for Data Grid", In proceeding of the 10th International conference on Parallel and Distributed Proceeding Techniques and Applications, IEEE Computer Society Washington, DC, WorldComp2010, GCA2010, LasVegas, USA, Volume3, pp: 95-100, July 12-15-2010.

[26] Cisco Distributed Director, http://www.cisco.com/warp/public/cc/pd/cxsr/dd/index.shtml

[27] M. Sayal, Y. Breitbart, P. Scheuermann, R. Vingralek, "Selection algorithms for replicated web servers". In Proceeding of the Workshop on Internet Server Performance,1998.

[28] E. Zegura, M. Ammar, Z. Fei, and S. Bhattacharjee, "Application-layer anycasting: a server selection architecture and use in a replicated web service", IEEE/ACM Transactions on Networking, vol. 8, no. 4, pp. 455–466, Aug. 2000.

.