Volume : 4 | Issue : 2 | February 2015 • ISSN No 2277 - 8179

Research Paper

Computer Science

KEYWORDS : Optimization technique; Data Grids; Replica Selection Technique; Data movement; Grids.

REPLICA OPTIMIZATION TECHNIQUE IN DATA GRIDS

Rafah M. Almuttairi | University of Babylon, Babylon, Iraq

ABSTRACT

Grid technologies are developed to be used for executing parallel applications over grid's sites in different geographical locations. For executing a grid job that contains a parallel application on a set of grid sites chosen by grid's broker, the files of data needed by the application are distributed in the data grid sites. The data grid files might be used subsequently by different grid's applications leading to multiple replicas of files in various data grid servers. Data grid files needed for a parallel application are transferred from the replica providers onto the computational site chosen by the grid broker for job execution. In this work, a novel optimization method is devised, to determine "set of cheapest" replica sites containing segments of the needed files. The objective of the proposed optimization technique, is to minimize the total transferring time needed for transferring the file's segments from the different replica sites to the computing site. Optimization technique is tested on different kinds of data grid experimental setups. We find that the best algorithm varies according to the configuration of replica providers, computational sites and clients.

Introduction

Data and computational grids are found to be powerful research-beds for executing different kinds of parallel and distributed applications [4]. To execute a parallel distributed application, broker or scheduler chooses a set of grid resources [5, 26]. In data grids, the required inputs data needed by the application are partitioned and distributed on the grid resources via data distribution strategy [28].

Grid's broker selects different sets of resources to execute parallel operations. Similarly, in high-energy physics experiments at CERN, the large amounts of data will be generated, same portions of the generated data might be used for various kinds of processing by different users/ application. Since the amount of data is large, some of the processing may involve parallel computations on the data whereby the data is partitioned and distributed among the resources used for parallel computations as shown in Figure 1.

In data grids, replication techniques are used for caching and replication policies, so that, parallel applications which are dealing with the same data can concurrently use set of data resources. Thus, replicas have to be created with different distributions on different sets of resources when the number of parallel computations on the same data is increased. Increasing replicas can help to reduce data transfer and access times for a computation and accordingly, various replica placement strategies have been proposed [21]. In this article, the following challenge that are related to replica selection are addressed :

- Select set of replica providers
- Selected replica providers share transferring job of the required files
- Cost of file transferring is reasonable
- Total file transferring time is low

The rest of the paper is organized as follows: section 2 is about the internal design of the replica broker, section 3 introduces the proposed Intelligent Optimizer highlights, section 4 covers simulation inputs and section 5 declares simulation results. Section 6 is for discussion conclusion.

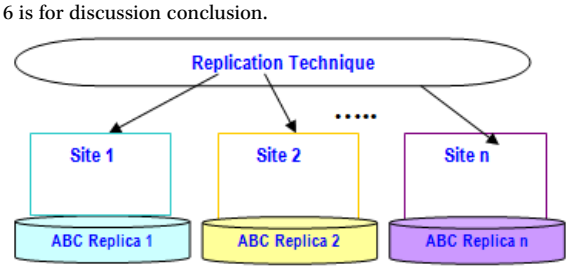

Figure1. Replication three files on different storage sites

Internal design of the Replica Broker

One of the most important operation of the previous sequenced operations is, selecting the best replicas using a Replica Broker. In this section the general description of our proposed Replica Broker is given. We followed the design of the GridBus Grid Service Broker [26] to design our Broker. In general the design of the Broker is composed of three main sub-systems:

User /Application interface sub-system

The interface layer is responsible of forwarding the input files which are:

A files description list:

It contains jobs need to be executed for users.

A replicas description list:

It contains current state of the available replica sites providers.

Core-sub-system

It converts the above inputs, lists of interface sub-system, into "jobs" and "replicas", however, "job" is the abstraction for a unit of work contains the names of the required files which are needed to be moved to a specific computational node, whereas, "replicas" contains the abstraction of replica providers.

The execution sub-system

Once the jobs are prepared and the replicas are discovered, the optimizer is started in the execution sub-system. As it is shown in Figure 2 below, the proposed optimizer selects a best replica provider(s) based on its selection algorithm. Execution sub-system has components which are:

Actuator component : it is a middleware specific component, dispatches the job to the remote storage of grid node.

Job-monitor : it updates the book-keeper by periodically monitoring the jobs using the services of the execution sub-system.

Job-manager : it takes care of cleaning up and gathering the output of the jobs when the job gets completed, Figure 2 illustrates the architecture of the internal design of GridBus Broker and shows the proposed Optimizer .

Volume : 4 | Issue : 2 | February 2015 • ISSN No 2277 - 8179

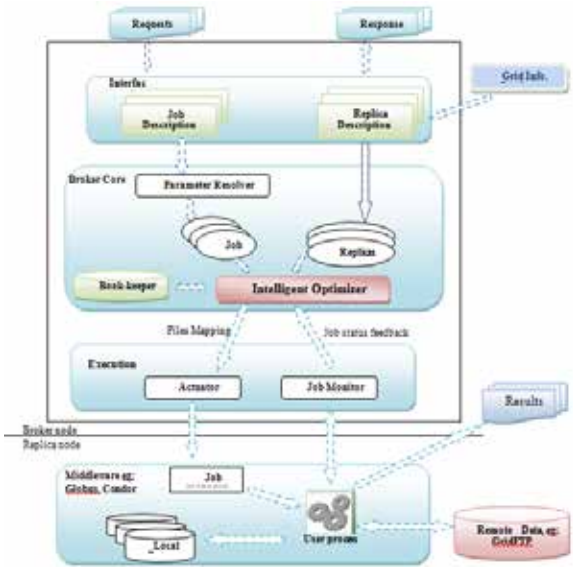

Figure 2: Enhanced Resource Broker architecture

Intelligent Optimizer

This section clarifies our approach, how it works, how it differs from the traditional model and what are the helpful advantages for us to cover the limitations of the traditional model. Our proposed optimizer uses two methods to optimize replica selection process.

discovering sets of associated replica sites

The concept of association rules of data mining approach is used to upgrade the capability of optimizer in the proposed selection technique. It means, the optimizer becomes able to choose multiple replica provider sites. The selected set of sites can concurrently share transferring files to minimize total transferring time which leads to speeding up executing data grid job as shown in Figure 3.

The selected set of replica sites should have similar characteristic of network condition, that is the main reason to use Pincer-Search algorithm. Before going ahead and explaining the steps of our proposed model, there is an important point of difference between the traditional model and our model that must be declared. In our model, we don't depend on the number of hops or the bandwidth criteria to select the best replica site. We use the stability of the network link as a criterion. It means the replica sites having the most stability links will be chosen even though their bandwidth or hops are not optimal. So the retransmission is going to be far less using our model than the traditional method.

To know the stability of links, we used a new testing route term called Single Trip Time (STT). STT is the time taken by the small packet to travel from replica's site to computing site. 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 computing site. It means that STT is the summation of all these delays which can reflect the stability of the links [14]. Before selection process starts, the computing site receives a periodically STTs of all replicas' sites and stores the most recent in a log file called Network History Database (NHD). To extract the best replicas from sites with the stable links, Pincer-Search algorithm is used. Pincer- Search algorithm is popular algorithm used for the association rules discovery to extract the hidden knowledge of the large data base.

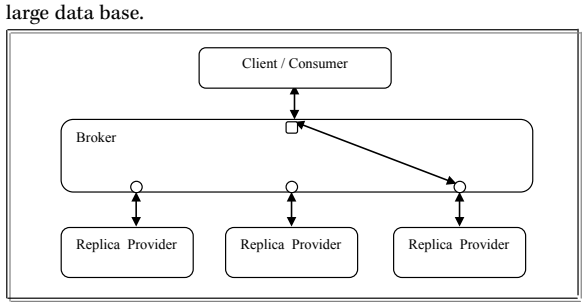

Figure 3: Multiple sites concurrently send different files

The notations and definitions of mining association rules of Pincer-Search algorithm as it has been introduced by Agrawal in 1993 [16] are introduced below:

Definitions:

• I: set of items, where I = {i1, i2, . . ., im}

• T: set of transactions, each transaction t is included in I. A transaction represents the values of RTTs between computing site and all replicas sites

• X: set of items from I

• $X \rightarrow Y$: An association rule, where $X \subseteq I$, $Y \subseteq I$, $X \cap Y = \emptyset$

• c: confidence of the rule $X \rightarrow Y$, if c% of the transactions in T that contain the set X, contain also the set Y with confidence c% of transactions in T

• s: support of rule X U Y, if s% of the transactions in T contains the set X U Y

• k : number of times of reading the whole database

• F: frequent Item with support s and minimum user support is s1, if $s \geq s1$

• Infrequent Item (E): an item which is not frequent is infrequent

• MFCS: Maximum Frequent Candidate Set

• MFS: Maximum Frequent item Set

• DGTT: Data Grid Transactions Table

• ES: Efficient Set

•NHF: Network History File (column represent the replicas sites and rows represent transactions

A Pincer-Search Method [5,8]

- Let $L_0 = \Phi$; $K=1$; $C1=\{\{i\}|i \in I\}$; $S0= \Phi$
- Let MFCS = {{1, 2, …, n}}; MFS = Φ
- Do until Ck= Φ and Sk-1= Φ
- Read DGTT and count support for Ck and MFCS
- MFS= MFS U {frequent items in MFCS}
- Sk= {infrequent items in Ck}
- call MSCS-gen procedure if Sk +1
- call MFS-pruning procedure
- generate candidates Ck+1 from Ck

- if any frequent item in Ck is removed in MFS-pruning procedure then,
-call the Recovery procedure to recover candidates to Ck+1
-call MFCS prune procedure to prune candidates in Ck+1
- k= K+1
- return Lk Uk

MFCS-gen procedure

- for all items s $\in$ Sk

- for s is a subset of m, MFCS = MFCS /{m}

- for all items e $\in$ s

- if m /{e} is not a subset of any item in MFCS then,

MFCS = MFCS U {m /{e}}

- ES= MFCS

- Return ES (it is output from Pincer-Search algorithm is a set of replicas which have a stable links called Efficient Set ES)

Recovery procedure [5]

- for all items c in C_k

- for all items m MFS

-if the first k-1 items in l are also in m /* suppose m.item = l.itemK-1 */

- for i from j+1 to |m|

Ck+1 = CK+1 U {{l.item1, l.item2, ….., l.itemk, m.itemi}}

MFS-Prune procedure [5]
- for all items c in C_k

- if c is a subset of any item in the current MFS then,

delete c from C_k

MFCS-Prune procedure [5]
- for all items c in C_{k+1}

- if c is not a subset of any item in the current MFCS then,

delete c from C_{k+1}

Hungarian algorithm for selecting cheapest replica set
The Hungarian algorithm is used to solve the linear assignment problem within time bounded by a polynomial expression of the number of agents. The assignment problem is a special case of the transportation problem, which is a special case of the minimum cost flow problem [15].

After getting set of sites with stable links by applying Pincer-Search algorithm, we need to assign a file send task for each replica site from this set of sites so that we get the least costs (prices). To do that, Hungarian algorithm is used.

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

ES Algorithm.

Step1: Data grid jobs are submitted by User/Resource Broker (RB) to RMS.

Step2: Gathers the replica location information from Replica Location Service (RLS)

Step3: By using Iperf service we probes the routes periodically between the computing site and all replicas sites to build the NHD

Column: The columns will represent the replicas sites; each column represents one site which has a copy of the requested file.
Row: The rows will represent the transactions; one transaction represents the values of STTs between computing site and all replica's sites.

Step 4: Calculate the threshold to change the NHD values to binary values

Threshold:
Calculate the means of each column $\overline{M = \sum_{i=1}^{N} Xi / N}$

Calculate the Standard deviation for each column as shown below:

$$Stdi = \sqrt{1/N \sum_{i=1}^{N} (X_i - M)^2}$$

X= RTT

N= No of transactions.
Find Q=(Stdi /M)*100
Calculate the Average of all Stdi of all columns. Av(Stdi)
Compare Av(Stdi) with Stdi
If (Stdi > Av(Stdi)) then make that value = 0 Otherwise 1
Step5: Apply Pincer-Search (PS)
PS (NHD,C,S,L) with following inputs:
Inputs: NHD: Network History Data Base which is built in step 4
C: Minimum confidence value.
S: Minimum support value.
Output: L: List of groups of sites.
Ln= U Gnm, n means the Group's order; m means the number of groups.
Step 6: ES $\subseteq$ Ln ,ES contains sites with stable links.
Step 7: Apply Hungarian algorithm (HA)
HA(ES, Co, HL):
Inputs: Two dimensions matrix with following inputs
Row: (ES) set of sites with stable links which is gained from step7.
Column: Costs of all files in all sites.
Output: Hungarian List of minimum costs, HL.

Step 8: Uses transport services such as GridFTP or UDT to transport the requested files.

Simulation inputs

ES approach is tested using:
The Network History File NHF Real of real Grid environment called *PRAGMA* Grid [7]. Uohyd nodes represent a Computing Site where the files should be moved to. The rest of sites represent the replicas where the required files are stored see Figure 4. *Ipref* service is used to get the history of Round trip time between *Uohyd* and other replica sites [4]

Cost of the replicas are taken using Grid Information Service that responsible to get information from replica providers

Volume : 4 | Issue : 2 | February 2015 • ISSN No 2277 - 8179

The Confidence *c* and Support *s* values are selected with respect to the nature of NHF data sets

Figure 4. PRAGMA Grid, 28 institutions in 17 regions [4]

To compare between our selection technique and traditional selection models an *OptorSim* [6] simulator is used. OptorSim is a Data Grid simulator designed to allow experiments and evaluations of various selection optimization strategies in Grid environments. Using OptorSim simulator, our program gets input from three configuration files

The Grid Configuration File

Grid Configuration File is used to describe the topology of the sites, their associated network geometry and the content of each site having resources available like number of worker nodes in computing element, the processing power per worker node, number of storage elements, the size in MB of storage element and network connections to other sites as shown in Figure 5. Each link between two sites shows the available network bandwidth. The network bandwidth is expressed in Mbits/sec (M) or Gbits/sec (G). The circles referred to the sites which are separated by stars referred to routers.

Job Configuration File prompt

This file contains the name of jobs, list of required files for each job, a list of logical file names, their sizes in MB and their unique numerical identifiers. It contains also Job selection probability and schedule table with a job execution time for each computing element.

Background Network Configuration File

This file is used to describe background network traffic. It is a site-by-site matrix, having for each pair of sites the name of data file containing the relevant RTT information, the Mean of RTT and the Standard Deviation; keeping the source sites in the rows, and the destination sites in the columns.

Initialization Configuration File

It is the last configuration file that initializes different parameters for running the simulation. These parameters may include information such as total number of jobs to be run, file processing time, delays between each job submission, maximum queue size in each computing element, file access pattern, the optimization algorithm used, etc. We assume a fixed replica lookup time, the time taken to lookup the catalog to get the physical locations of replica's sites which is equal to 20 seconds. After transferring file process is finished, the file index is recorded, with the Mean and Standard Deviation of STT into a log file NHD. Using Hungarian model the set of cheapest files prices sites will be chosen as an Efficient Set (ES) to get requested files from. As mentioned above, a job will typically request a set of logical filename(s) for data access. The order in which the files are requested is determined by the access pattern. In data grid simulation, there are four access patterns: sequential (files are accessed in the order that has been stated in the job configuration file), random (files are accessed using flat random distribution), unitary random (file requests are one element away from previous file requests but the direction will be random), and Gaussian random walk (files are accessed using a Gaussian distribution).

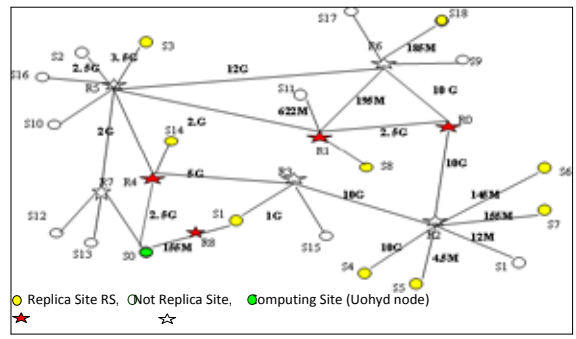

Figure 5. PRAGMA Grid and their associated network geometry, S0 represent UoHyd [7]

In our simulation run, we proposed all replicas that have the same characteristics like: number of jobs to be run, file processing time, delays between each job submission, maximum queue size in each computing element, file access pattern size and number of requested files and speed of input/output storage operations. We applied the similar with two different replica selection models:

Using the Traditional Model with different options (Hop counts, Bandwidth and Round Trip Time)

Using Efficient Set Model ESM (our proposed model), to see the affect of congested links of both models

Simulation result

The simulator called, OptorSim is used to see the difference in the result when we use the previous models and our proposed model as a replica selection technique and compare between the total file transfer times and cost of data grid job.

5.1 A Comparison of the file transmission time of requested file using Neural Network Model (NN) and Efficient Sites Model (ESM)

In this section, we will analyze the results using the neural network model NN and efficient site model ESM. As we mentioned before (in the simulation inputs section) that all configuration parameters are the same for all replica sites. The only difference is the network conditions (Number of counts, BW, RTT and prices of files).

As it is seen in Figure 6, when the replica selection Broker uses ESM as an optimizer, the total files transferring time will be shorter than using prediction model such as NN or regression models [17].

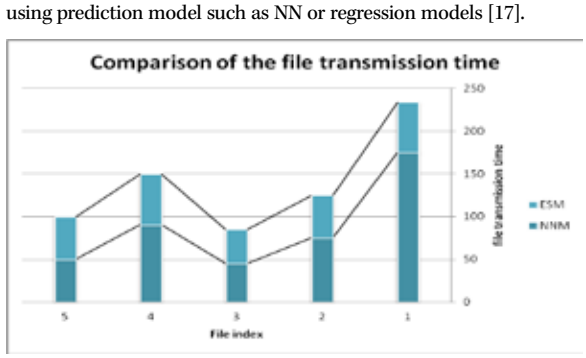

Figure 6. Comparison of the file transmission time between ESM and NN

5.4 A Comparison of Costs (prices) of files using Efficient Sites Model with Random and Sequential selec-

TION MODELS

In this section, a comparison between ESM and Random Model is explained. Both models, ESM and RM are used to select the cheapest replica sites. Let us use a study case, the list of required files is $\{f_1, f_2, f_3, f_4, f_5\}$, and the list of selected replicas using ES is ES= $\{S_1, S_2, S_3, S_4, S_5\}$, so five sites are needed to concurrently transfer five files in case one site sends a single file.

Random Model

In this model, the five sites are selected randomly to get the required files of J_1 [28].

So, Random Selection List is, SL={$f_1 \rightarrow S_1$, $f_2 \rightarrow S_2$, $f_3 \rightarrow S_5$, $f_4 \rightarrow S_4$, $f_5 \rightarrow S_3$}. Now to find list of files using ESM when Hungarian method is applied. The result of selection is: HL= {$f_1 \rightarrow S_5$, $f_2 \rightarrow S_1$, $f_3 \rightarrow S_3$, $f_4 \rightarrow S_2$, $f_5 \rightarrow S_4$}. Then, we compared the total price of files of the Random Model with the price of the files selected using ESM As shown in Figure 7 below the price of the files using ESM is $89 whereas the price of the same files using Random Modem is $112 [28].

Sequential Model SM

SM is another selection model is used to compare the result with our proposed model ESM. The selection of the Sequential Model is done as follow [28]:

Sequential selection model is SL= {$f_1 \rightarrow S_1$, $f_2 \rightarrow S_2$, $f_3 \rightarrow S_3$, $f_4 \rightarrow S_4$, $f_5 \rightarrow S_5$}, the total price of the Sequential Model is $101 whereas the total price of ESM is $89. Our proposed model ESM uses the Hungarian algorithm always gives better way to get files from replica providers with cheap price as shown in Figure 7.

Figure 7. Comparison of three selection methods

6. Discussion and conclusion

In this paper, we proposed a new replica selection model in data grid to optimize the following points:

- Minimizing the total time of executing the job by minimizing file transfer time
- Minimizing the total cost of files
- Our model utilizes two algorithms
- Pincer-Search algorithm for first optimization point [8]
- Hungarian algorithm for second optimization point [15]

The difference between our model and the traditional model is:

Our technique gets a set of sites with stable links work concurrently to transfer requested files. The traditional model selects one site as a best replica's site and getting a set of sites would not reflect the real network condition. i.e., most probably this model will not pay any attention whether these sites uncongested links or not at the transferring moment because the traditional model depends upon the Bandwidth alone or Hop counts alone which do not describe the real network condition, whereas we depend on the STT which reflects the real network conditions.

7. Future work

Being a node of PRAGMA we are looking forward to deploy our technique as an independent service in PRAGMA data grid infrastructure to speed up the execution of data grid job and minimize total cost of requested files.

REFERENCE

[1] Rashedur M. 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] 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). | [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] http://goc.pragma-grid.net/cgi-bin/scmsweb/probe.cgi?source=GOC&grid=PRAGMA | [5] A. K. PujarL, "Data Mining Techniques". Niversities Press, India, 2001. | [6] W. Bell, et al., "OptorSim - A grid simulator for studying dynamic data replication strategies", Journal of High Performance Computing Applications 17 (4) (2003) | [7] http://goc.pragma-grid.net/wiki/index.php/UoHyd | [8] DI Lin, ZM Kedem, "Pincer-Search: A New Algorithm for Discovering the Maximum Frequent Set", - IEEE Transactions on Knowledge and Data, 2002. | [9] S. Vazhkudai, S. Tuecke, I. Foster, "Replica Selection in the Globus Data Grid", Cluster Computing and the Grid, IEEE International Symposium on, p. 106, 1st IEEE International Symposium on Cluster Computing and the Grid (CCGrid'01), 2001 | [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] I. Foster, S. Vazhkudai, and J. Schopf. " Predicting the performance of wide-area data transfers". In Proceedings of the 2008 ACM/IEEE conference on Supercomputing (SC 2008), pages 286-297, Nov. 2008. | [14] J. F. Kurose and K. W. Ross. "Computer networking - A top-down approach featuring the internet". Addison-Wesley-Longman, 2001. | [15] http://en.wikipedia.org/wiki/Assignment-problem | [16] R. Agrawal, I. Tomasz, and A. Swami. "Mining association rules between sets of items in large databases". In Proceedings of the 1993 ACM SIGMOD international conference on Management of data, SIGMOD '93, pages 207-216, New York, NY, USA, 1993. ACM. | [17] R. M. Rahman, R. Alhajj, and K. Barker. "Replica selection strategies in data grid". In Journal of Parallel Distributing and Computing, volume 68, pages 1561-1574, 2008. | [18] A. Jaradat, R. Salleh, and A. Abid. "Imitating k-means to enhance data selection". In Journal of Applied Sciences 2009, volume 19, pages 3569-3574, 2009. | [19] R. M. Almuttairi, R. Wankar, A. Negi, C. R. Rao, and M. S. Almahna., "New replica selection technique for binding replica sites in data grids". In Proceedings of 1st International Conference on Energy, Power and Control (EPC-IQ), pages 187-194, Dec. 2010. | [20] R. M. Almuttairi, R. Wankar, A. Negi, and C. R. Rao., "Intelligent replica selection strategy for data grid". In Proceedings of the 2010 International Conference on Grid Computing and Applications (GCA'10), pages 95-101, 2010. | [21] R. M. Almuttairi, R. Wankar, A. Negi, and C. R. Rao., "Replica selection in data grids using preconditioning of decision attributes by k-means clustering". In Proceedings of the Vaagdevi International Conference on Information Technology for Real World Problems, pages 18-23, Los Alamitos, CA, USA, 2010. IEEE Computer Society. | [22] R. M. Almuttairi, R. Wankar, A. Negi, and C. R. Rao., "Rough set clustering approach to replica selection in data grids". In Proceedings of the 10th International Conference on Intelligent Systems Design and Applications (ISDA), pages 1195-1200, Egypt, 2010. | [23] R. M. Almuttairi, R. Wankar, A. Negi, and C. R. Rao., "Smart replica selection for data grids using rough set approximations". In Proceedings of the International Conference on Computational Intelligence and Communication Networks (CICN2010), pages 466-471, Los Alamitos, CA, USA, 2010. IEEE Computer Society. | [24] R. M. Almuttairi, R. Wankar, A. Negi, and C. R. Rao., "Enhanced data replication broker". In Proceedings of the 5th Multi-disciplinary International Workshop on Artificial Intelligence (MIWAI2011), pages 286{297, 2011. | [25] Praveen Ganghishetti, Rajeev Wankar, Rafah M. Almuttairi, C. Raghavendra Rao., "Rough Set Based Quality of Service Design for Service Provisioning in Clouds", 6th International Conference on Rough Sets and Knowledge Technology (RSKT 2011) pp. 268-273, Lecture Notes in Computer Science 6954, Springer, 2011, ISBN 978-3-642-24424-7, October 2011, Canada. | [26] K. Nadiminti, S. Venugopal, H. Gibbins, T. Ma, and R. Buyya. The grid-bus grid service broker and scheduler (v.2.4) user guide. In online at: http://www.Gridbus.org/broker/2.4/manualv2.4.pdf, 2005. | [27] S. Vadhiyar, J. Dongarra, GrADSolve , A grid-based rpc system for parallel computing with application-level scheduling, Journal of Parallel and Distributed Computing 64 (2004) 774–783. | [28] M. S. Almahanaa, R. M. Almuttairi, "Enhanced Replica Selection Technique for binding Replica Sites in Data Grids", In Pro. Of International Conference on Intelligent Infrastructure, 47th Annual National Convention of the Computer Society of India organized by The Kolkata Chapter December 1-2, 2012, Science City, Kolkata. |