NEW REPLICA SELECTION TECHNIQUE FOR BINDING CHEAPEST REPLICA SITES IN DATA GRIDS

Rafah M. Almuttairi

Department of Computer Science, University of Babylon, P.O. Box, 4, Babylon City, Iraq
rafahmohammed@gmail.com

ABSTRACT

Data Grid technology is developed to share data across many sites in different geographical locations primarily to improve data access and transfer performance. When different sites hold a copy of the files, there are significant benefits realized when selecting the best set of sites that cooperate to speed up the file transfer process. In this paper, we describe a new optimization technique used to optimize two things when a new transfer replica file is requested. First it is minimizing total file transferring time by applying a Pincer-Search algorithm of Data mining approach to discover associated distributed replicas sites to share the transferring file process. Second, it is minimizing the cost (price) of requested file(s) by using Hungarian algorithm.

KEYWORDS

Pincer-Search algorithm, Data Replica Service, Selection Technique & Hungarian algorithm.

1. INTRODUCTION

Grid computing emerges from the need to integrate collection of distributed computing resources to offer performance unattainable by any single machine. Grid technology facilitates data sharing across many organizations in different geographical locations. Data replication is an excellent technique to move and cache data close to users. Replication is a solution for many grid-based applications such as climatic data analysis and Grid Physics Network [6] which both require responsive navigation and manipulation of large-scale datasets. Moreover, if multiple replicas exist, a replica management service is required to discover the available replicas and select the best replica that matches the user's quality of service requirements. The replica selection problem can be divided into two sub-problems:

1) Discover the physical location(s) of a file, given a logical file name; and

2) Select the best replica from a set based on certain selection criterion [4].

Since the same file has multiple physical names in different storage locations, the job of Replica Location Service (RLS) is to maintain associations or mappings between logical file names and one or more physical file names of replicas. A data grid job can provide a logical file name to a RLS server and ask for all the registered physical file names of replicas as shown in Figure 1.

Natarajan Meghanathan, et al. (Eds): SIPM, FCST, ITCA, WSE, ACSIT, CS & IT 06, pp. 295–312, 2012.
 DOI : 10.5121/csit.2012.2329

Data Grid Job is a job that consists of the name of group of files which are asked by user/application to do different analyzing. All data grid jobs are submitted to Data Management System DMS. DMS queries the Replica Location Service to find out whether there is a local copy of the file. If not, a list of file existence locations is sent to DMS. Then, DMS generates a request to copy the file to the local storage system and registers the new copy in the local RLS server. Since more than one replica of the requested file is available, the Replica Selection becomes an important decision because it effects on the efficiency of executing data grid job [9]. Traditional Model TM selects the best replica site by probing the links between computing site (the site in which the job is executed) and all replicas sites (sites which have a copy of the requested file). The site with maximum bandwidth link or least Hop count will be selected as a best replica site to send the requested file. TM does not give an optimal selection in two cases:

a) When the links are congested, since the Bandwidth and Hop count do not reflect the real condition of links
b) When the file has different cost (price of the file). TM does not select the cheapest site. The selected site using TM might not be the cheapest site

To cover both the limitations, we proposed a new replica selection technique that selects the best set of replicas. It improves the selection performance of TM. The new selection technique covers the first limitation using association rules of Data mining approach. Pincer-Search algorithm is used to discover the uncongested links. The second limitation has been covered by using Hungarian algorithm. Hungarian algorithm is used as an optimizer that selects the associated sites with a minimum cost (price). The rest of the paper is organized as follows: section 2 is about the related works, section 3 highlights the common aspects of replication architecture, section 4 introduces replica selection models, section 5 covers simulation inputs and section 6 declares simulation results. Section 5 covers the analysis of our technique and last two sections give a conclusion and scope for future work.

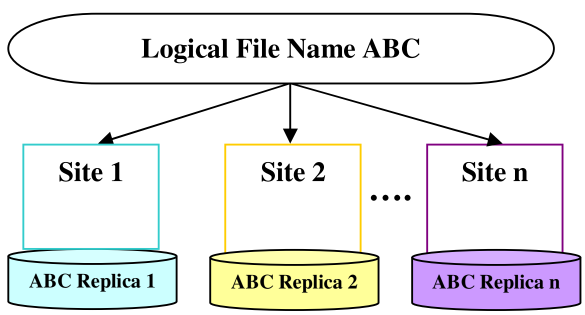


Figure 1. Example of the associations between a logical file name and three replicas on different storage sites

2. RELATED WORKS

The replica selecting problem has been investigated by many researchers. Some of their works will be explained below to know what they have proposed and where exactly our work differs from the others. In 2001, Kavitha et al. [3], they used traditional replica catalog based model, where for each new request Replica Location Service will be 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 like Network Bandwidth and link’s latency. From 2001-2003, Sudharshan et al [2], In this work they used the history of previous file transfers information to predict the best site holing copy of requested file. When a file transfer has been made between two sites, the file size, the available network bandwidth, and transfer time are saved so it can be used later for training and testing the regression model to predict the transfer time. In this work they show 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], in this work a replica selection technique called the K-Nearest Neighbor (KNN) rule is exploited to select the best replica from 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 nearby. This technique has a drawback as they mentioned in their paper: the misclassification will increased when for large file transfer and costs more than a couple of small file transfer misclassifications. Especially in the Gaussian random access pattern the accuracy is the lowest. Another drawback is KNN needs to save all previous instances (requests of files) to use them to select the best replica's site which means it will take a 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 predictive technique (NN) to estimate the transfer time between sites. The predicted transfer time can be used as an estimate to select the best replica's site among different sites. Simulation results demonstrate that Neural Network predictive technique more accurately than the multi-regression model, which was used before NN [2]. However NN technique does not always give the right decision because the copy of the file may become no longer available (this is common in grid environment because the memory of site is limited) in the predicted site which has the lowest transfer time, so in this case they will have to use traditional model. 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 transferred. In our previous works [19, 20, 21, 22, 23, 24, 25], we enhanced the selection strategy using Rough Set theory [21, 22, 25] and Apriori algorithm [19, 20, 24]. Our current work differs from the previous works by selecting not only one replica site, but number of sites that have similar characteristics in terms of stabilizing the network conditions. Pincer-Search algorithm is better than Apriori algorithm as it consumes less searching time of getting a stable set of replicas. These sites concurrently work to send parts of a big file or different small files with the cheapest cost.

3. DATA GRID ARCHITECTURE

In this section, Data Grid Architecture is explained with functionality of each component.

3.1. Replica Management System (RMS)

As we see in Figure 2 below, the main component of Data Grid is RMS. RMS acts as a logical single entry point to the system and interacts with the other components of the systems as follows:

3.2. Replica Location Service (RLS)

Replica Location Service (RLS) is the service that keeps track of where replicas exist on physical storage systems and 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 detail, we need to define a few terms.

- A Logical File Name (LFN) is a unique identifier for the contents of a file
- A Physical File Name (PFN) 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 [9].

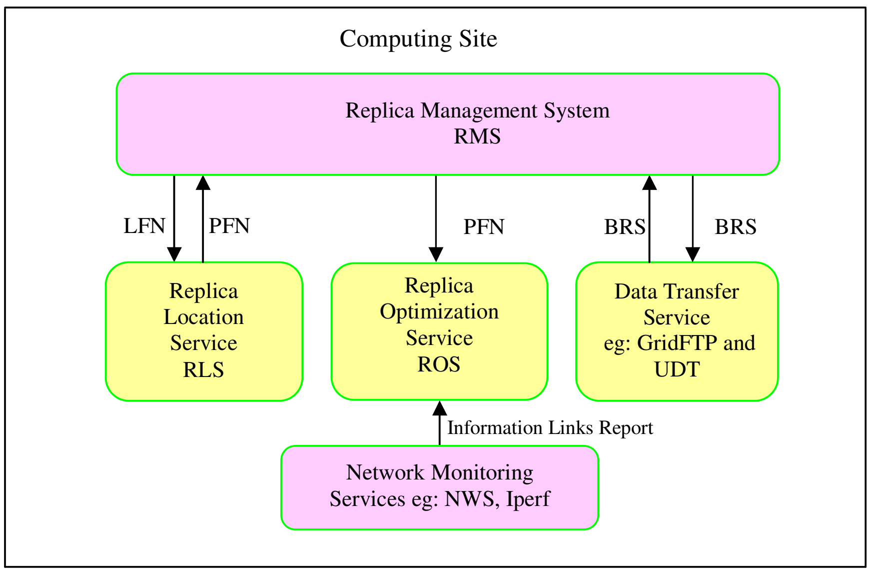


Figure 2. Functionality of Replica Management System

3.3. Replica Optimization Service (ROS)

The Optimization component is used to minimize the time of file transfer by pointing access requests to appropriate replicas. It also replicates the 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 network and storage access latencies [4]. ROS gathers 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.

3.4. 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 such as GridFTP [13] and UDT [12]. After getting a simple and a clear picture about the infrastructure of data grid, next section explains where our model resides and how it (our model) changes the data grid performance.

4. REPLICA SELECTION STRATEGIES

As we mentioned before, a replica optimization component of data grid is responsible to use a selection strategy to select the best site holding a copy of requested file(s) at the time of executing the job. Generally traditional model is used to do the selection process. In this paper, we proposed a new model called Efficient Replica Set ERS model. The following subsections are used to illustrate the TM and EST models.

4.1. Traditional Selection Model (TM)

Whenever the new job is submitted to RMS and the requested file(s) is unavailable in the local storage element, RMS queries ROS to select the best site(s) holding a copy of the requested file(s). In traditional model the selecting process depends on different criteria like selecting the best replica site which has the fastest input/output memory operations, selecting the best replica site which has fewer queues waiting requests length and selecting the best replica site depending on network condition. The Network Condition selection criterion includes the following steps:

- Probing the links between the Computing Site (CS) and all replicas sites
- Counting the number Hops (number of routers between two parties) or checking the maximum network bandwidth between them. Depending on network condition of the links, the best replica site will be the one which has either less Hop counts or maximum bandwidth value of the network link between two parties

4.2. Efficient Replica Set Selection Model (ESM)

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 technique uses two methods to optimize replica selection process. The following subsections are to declare both in details.

4.2.1 Pincer- Search algorithm for discovering sets of associated replica sites

Our aim of using Pincer-Search algorithm is to select more than one replica site working concurrently to minimize total time of transferring the requested file(s) which means speeding up executing data grid job as shown in Figure 3.

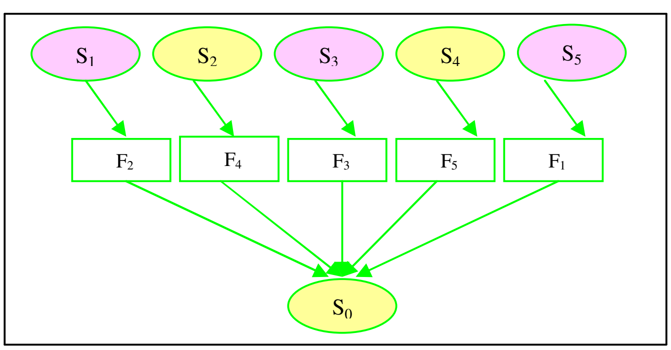


Figure 3. Multiple sites concurrently send different files

The core idea is, the selected set of replica sites have *similar characteristic of network condition*. To do that the association rules concept of Data mining approach is used. The selected sites have uncongested links (stable links). Required data moves fast in the stable links so the total transfer time is minimized. In case that files have a different prices, the selection strategy has to be efficient enough to select most of the required files from the cheapest sites. In our work we a Pincer-Search algorithm is used for this purpose. The sequence steps of getting set of stable site using Pincer-Search algorithm is cleared in Figure 4.

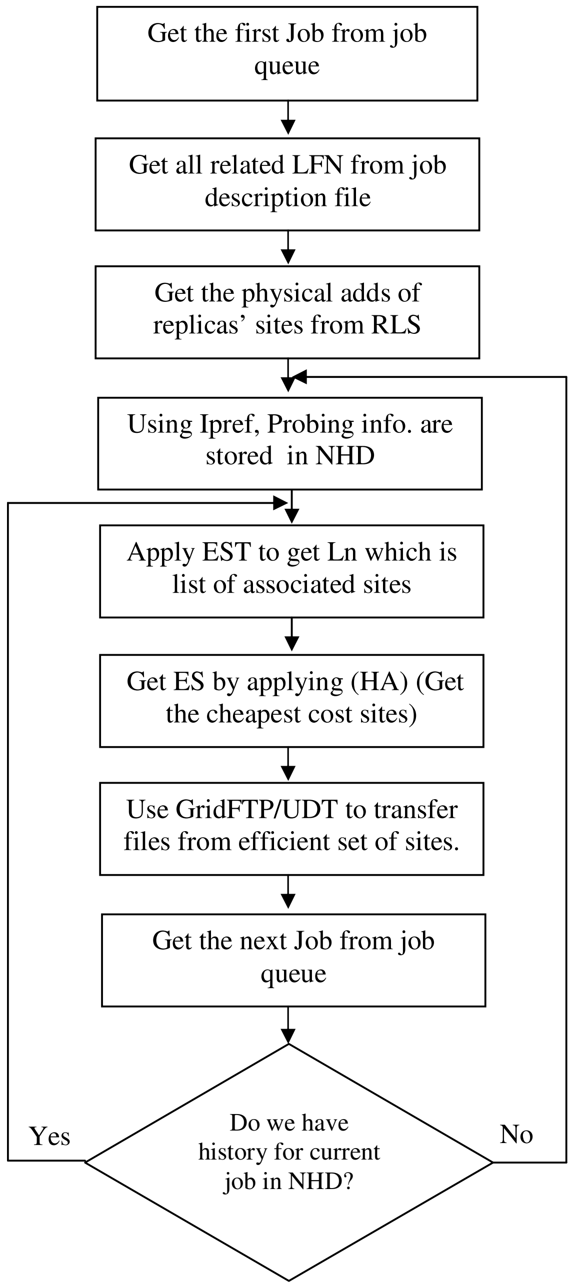


Figure 4. Execution flows using ESM

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 $I = \{i_1, i_2, \ldots, i_m\}$
- 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 $X \cup Y$, if s% of the transactions in T contains the set $X \cup Y$
- k : number of times of reading the whole database
- F: frequent Item with support *s* and minimum user support is s_1, if $s \geq s_1$
- 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 $L_0 = \Phi$; $K=1$; $C_1=\{\{i\}|i \in I\}$; $S_0= \Phi$
- Let $MFCS = \{\{1, 2, \ldots, n\}\}; MFS = \Phi$
- Do until $C_k= \Phi$ and $S_{k-1}= \Phi$
  - Read *DGTT* and count support for C_k and *MFCS*
  - MFS= MFS U {frequent items in MFCS}
  - S_k= {infrequent items in C_k}
  - call MSCS-gen procedure if $S_k \neq \Phi$
  - call MFS-pruning procedure
  - generate candidates C_{k+1} from C_k
  - if any frequent item in C_k is removed in MFS-pruning procedure then,
    - call the Recovery procedure to recover candidates to C_{k+1}
    - call *MFCS* prune procedure to prune candidates in C_{k+1}
  - $k= K+1$
  - return L_k U_k

MFCS-gen procedure

- for all items $s \in S_k$
- for s is a subset of m, $MFCS = MFCS \backslash \{m\}$
- for all items e $\in$ s
  - if $m \backslash \{e\}$ is not a subset of any item in *MFCS* then,
    $MFCS = MFCS\ U\ \{m \backslash \{e\}\}$
- $ES= MFCS$
- 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 $l \in C_k$
- for all items $m \in MFS$
  - if the first $k\text{-}1$ items in l are also in m
    /* suppose $m.item = l.item_{K-1}$ */
- for i from $j+1$ to $|m|$
  $C_{k+1} = C_{K+1}\ U\ \{\{l.item_1, l.item_2, \ldots, l.item_k, m.item_i\}\}$

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}

4.2.2 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 (replica provider sites).

The list of required files and the set of stable replicas (output of Pincer-Search algorithm ES) are used as input parameters to Hungarian algorithm.

To minimize the total cost (price) of getting the required files, each file or part of a file is taken from one site of *ES*. The decision of assignment files to sites is done using Hungarian algorithm. To get a clear idea let us go through the following example.

Study case:

Suppose the data grid job is submitted to RMS asking to get five logical names: (f_1, f_2, f_3, f_4, f_5). And suppose after applying Pincer-Search algorithm we get a set of sites in ES with different prices. For example: the cost of f_1 is $20 if we get it from S_1 but, it costs $18 if we get it from S_4. So to get the requested files at the same time from multiple replica sites with minimum cost (price), Hungarian algorithm is applied [15]:

Input parameters:

- List of required files: {f_1, f_2, f_3, f_4, f_5}
- ES= {S_1, S_2, S_3, S_4, S_5}.
- The cost (price) of each file from each site as shown in Table1

Table 1. Actual cost of each file in different sites.

	S_1	S_2	S_3	S_4	S_5
f_1	$20	$22	$25	$22	$18
f_2	$11	$26	$24	$24	$21
f_3	$23	$24	$17	$19	$18
f_4	$22	$20	$21	$23	$20
f_5	$18	$23	$25	$21	$25

Processing

Hungarian algorithm is applied on Table 1 to get the minimum cost it can be paid to get all requested files in a list called Hungarian List (HL) as shown in Table 2 (below).

Table 2. Files and cheapest providers map.

	S_1	S_2	S_3	S_4	S_5
f_1	**2**	**4**	**6**	**1**	**0**
f_2	**0**	**9**	**6**	**4**	**4**
f_3	**7**	**8**	**0**	**0**	**3**
f_4	**2**	**0**	**0**	**0**	**0**
f_5	**0**	**5**	**6**	**0**	**7**

Table2 shows a map for assignment decision. The optimal files and sites assignment is given below:

HL= {$f_1 \rightarrow S_5$, $f_2 \rightarrow S_1$, $f_3 \rightarrow S_3$, $f_4 \rightarrow S_2$, $f_5 \rightarrow S_4$}

It means the minimum cost will be achieved if , S_5 sends f_1, S_1 sends f_2, S_3 send f_3, S_2 sends f_4, and S_4 sends f_5. In our example the minimum cost to get all five files is: $87.

4.2.3 Efficient Set algorithm ES

Efficient Set algorithm is used to get the best set of replica sites working concurrently with minimum cost of getting the requested files. In this section, the steps of our proposed algorithm ES are explained.

ES algorithm steps:

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

Step2: Contact Replica Location Service (RLS) to get replica location information such as PFNs and cost of files

Step3: Contact Network Monitoring service such as Iperf or NWS to get NHF (it probes the routes periodically between the computing site and all replicas sites to build the NHF)

Step4: Apply Pincer-Search PS

PS (input: NHF, c, s; output: ES) with following inputs:

Inputs:

NHF: Network History File

c: Minimum confidence value

s: Minimum support value

Output: ES list of stable replicas sites

Step5: Apply Hungarian algorithm (HA) as shown in Table 1 and 2 above

HA (input: ES, Co; output: HL)

Inputs:

ES: set of stable replicas sites gained from Step 4

Co: Costs of all files in all sites

Output:

HL: Hungarian List of minimum costs

Step 6: Contact transport services such as GridFTP or UDT to move the requested files from the selected site ES with respect of HL

5. SIMULATION INPUTS

ES approach is tested using:

1- 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 5. *Ipref* service is used to get the history of Round trip time between *Uohyd* and other replica sites [4]
2- Cost of the replicas are taken using Grid Information Service that responsible to get information from replica providers
3- The Confidence *c* and Support *s* values are selected with respect to the nature of NHF data sets

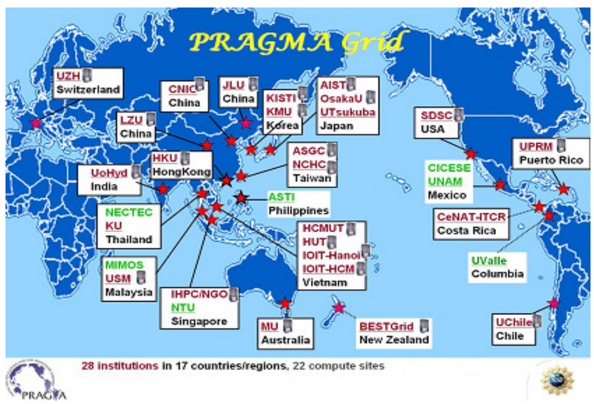


Figure 5. PRAGMA Grid, 28 institutions in 17

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

5.1 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 6. 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.

5.2 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.

5.3 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.

5.4 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).

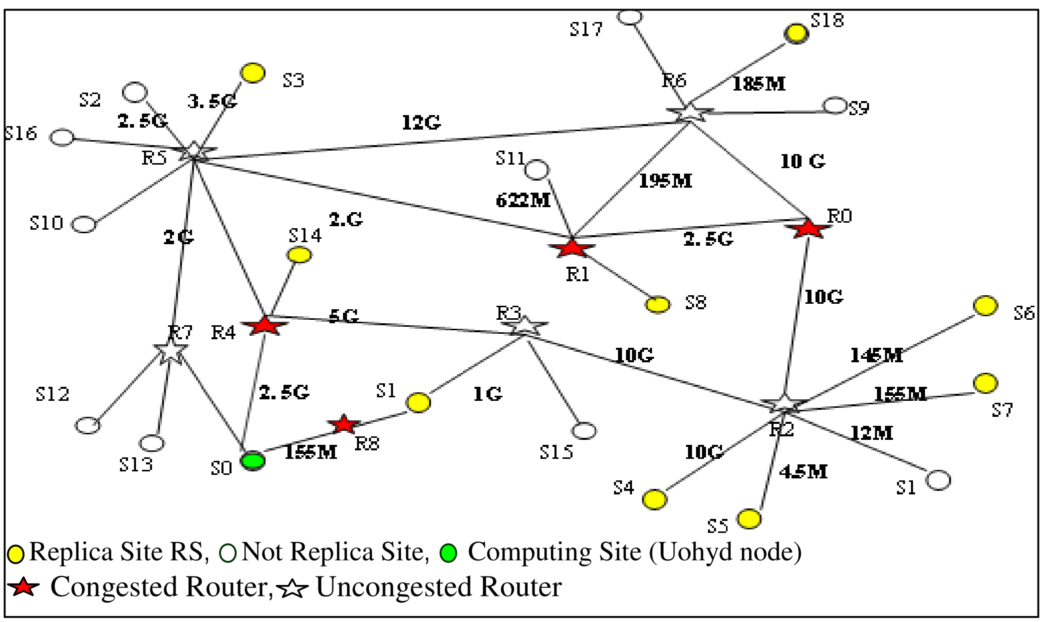


Figure 6. PRAGMA Grid and their associated network geometry, S_0 represent UoHyd [7]

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:

1- Using the Traditional Model with different options (Hop counts, Bandwidth and Round Trip Time)
2- Using Efficient Set Model ESM (our proposed model), to see the affect of congested links of both models

6. SIMULATION RESULT

As we mentioned before, we used OptorSim to see the difference in the result when we use the traditional model and our proposed model as a replica selection technique and compare between the total file transfer times of data grid job.

6.1 Execution flows of replica selection by using traditional model in the simulator

As we discussed earlier, the computing elements need to access files described in the job. If the file is in the local site, the computing element can access it instantly. Otherwise, the file must be transferred from the storage site that has the minimum file transfer time to the computing site. In the traditional model, the site gets the best replica of a file in the following way: it contacts the replica manager with a logical file name; the replica manager in turn calls the local replica catalog to determine the physical locations of the logical file. Once it finds the physical locations for the logical file, it gets the current network conditions (Bandwidth, number of Hop count or Round Trip Time (RTT)), the replica site which has the maximum bandwidth, the least number of hop counts or the least RTT is considered as the best site to fetch the file from.

6.2 Execution flows of replica selection algorithm by using ESM in the simulator

The execution of a computing element for the ESM is illustrated in Figure 4, the flow shows that if a computing element has a sufficient scanning history of links of the replica sites, it finds a list of efficient sites using the Efficient Sites Technique rule, otherwise it contacts the Iperf or NWS services to get a real view of routes. Initially, each computing element gets information from the replica catalog about the replica's sites.

The next section presents the simulation results and analyzes the performance of the replica selection algorithms with different performance metrics.

6.3 A Comparison of the file transmission time of requested file using Traditional Model and Efficient Sites Model

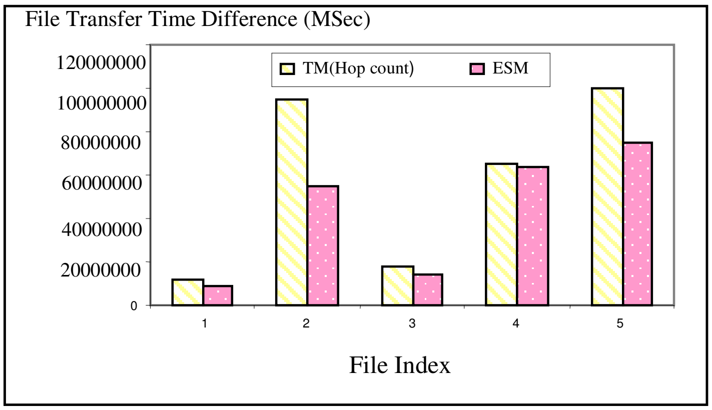
In this section, we will analyze the results using the traditional model with all different options and efficient model. As we mentioned before (in the simulation inputs section) that all configuration parameters are the same for all replica sites. The only difference is the network conditions (Number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

A) Traditional Model (using Hops as criteria to select best replica site) vs. ESM

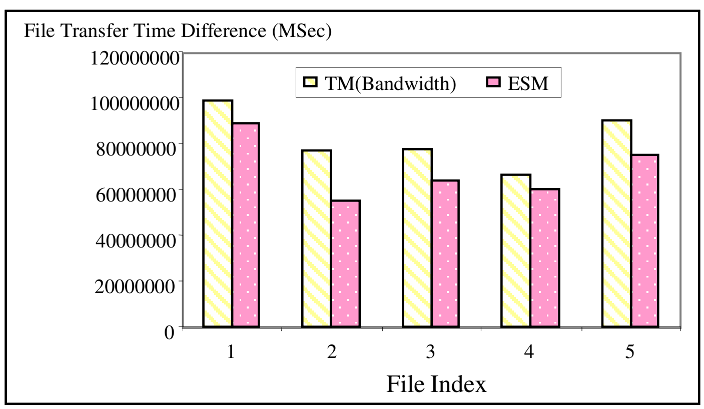
As we see below in Figure 7-a, the ES model is more efficient than the traditional model. The main reason is the ESM selects the best replica from the sites with the stable links. So the retransmission is far less than traditional method which selects the best replica from the sites having the least number of Hop Counts which do not reflect the real picture of the network, especially in the Inter-Grid architecture. Let us take an example to make it clearer. In Figure 6, If the job J_1 submitting to S_0 needs file resides in (S_1, S_3, S_4, S_{14}).Using the Traditional Model with a less number of Hop count S_0 will ask S_1 or S_{14} to get the file from because the number of Hops is one whereas the number of Hops between S_0 and S_3 is 2 and between S_0 and S_4 is 3. The time of file transfer in this case will be larger than the time of transfer the file from S_3 because the two routers (R_5 and R_7) are uncongested whereas R_4 is congested. Same thing will happen when TM chooses the best site depending on the highest bandwidth between two sites.

B) Traditional Model (using BW as criteria to select best replica site) vs. ESM

As we see in Figure 7-b) (below), ESM is better than TM. The traditional model here selects the best replica site depending on the bandwidth between the computing site and replica's site. The site having the maximum bandwidth will be chosen as a best replica site. The bandwidth of the link does not reflect the real picture of the network link separating two sites. It means that the link with less value of bandwidth might be faster.



(a) Traditional Model (TM) using Hop



(b) Traditional Model (TM) using the Bandwidth

Figure 7. Transferring time of requested files by Job1 using traditional model and Efficient Set Model (ESM)

C) Traditional Model (using RTT as criteria to select best replica site) vs. ESM

In this section, the traditional model selects the best replica site depending on the Round Trip Time value. RTT (Round Trip Time) is the time taken by the small packet to travel from a client to server and back to the client. The RTT delays include packet-transmission delays (the transmission rate out of each router and out of the source host), 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 source host) for two directions (from computing site to replica site and vice versa). Therefore, the total delays from one end to another will be the summation of all these delays [14]. Even though RTT will reflect the real picture of network link, but still ESM is better because it probes one direction route. It looks to a single trip starting from replica site to computing site because this direction will affect on transferring the files but not the other once. We test ESM with the least RTT, the least Means of RTT and the least Standard Deviation of RTT values of RTT as shown in Figure 8.

As we see the ESM is better than all RTT values criterion

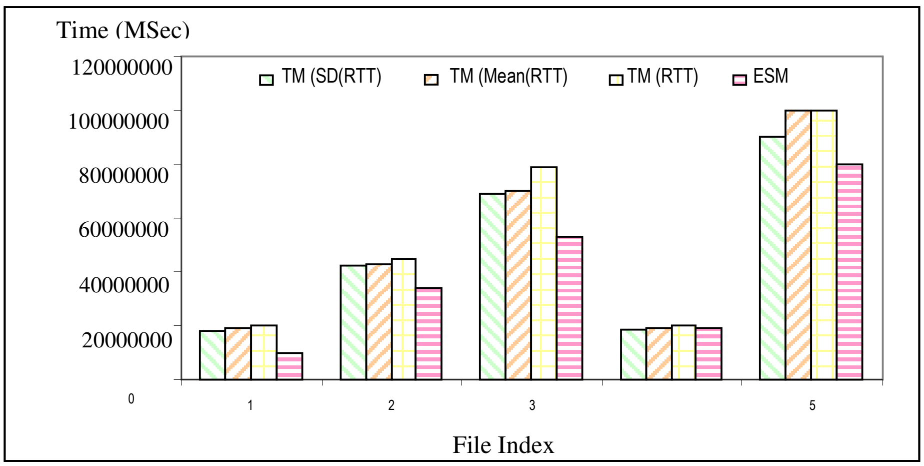


Figure 8. Transferring time of requested files by Job1 using TM using (RTT) and Efficient Set Model (ESM)

6.4 A Comparison of Costs (prices) of files using Efficient Sites Model with Random and Sequential selection 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 the study case of *Section 4.2.2***,** 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.

A) Random Model

In this model, the five sites are selected randomly to get the required files of J_1.
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 Hungarian method is applied. The result of selection using ESM 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 9 below the price of the files using ESM is \$89 whereas the price of the same files using Random Modem is \$112.

A) 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:

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 that uses the Hungarian algorithm always gives better way to get files from replica providers with cheap price as shown in Figure 9.

Figure 9. Comparison of three selection methods

7. DISCUSSION AND CONCLUSION

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

1. Minimizing the total time of executing the job by minimizing file transfer time
2. Minimizing the total cost of files

Our model utilizes two algorithms

1. Pincer-Search algorithm for first optimization point [8]
2. 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.

8. 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.

REFERENCES

[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.

Author

Rafah M. almuttairi is currently a faculty member at Babylon University, Babylon, Iraq. She has completed her Ph. D. research in University of Hyderabad, India in April, 2012. She received her Master Degree in **Computer** Science from Baghdad University, Iraq in October, 2003. She has received her Bachelor Degree in **Computer** Science and Bachelor Degree in **Physics** Science from Babylon University, Iraq in October, 2001 and October 1994 respectively. She has authored several international conferences papers in area of grid computing. Her current research interest is in the area of data grid architecture and fuzzy decision making for grid resources and replica selection.