Enhanced Data Replication Broker

Rafah M. Almuttairi, Rajeev Wankar, Atul Negi,
and Chillarige Raghavendra Rao

University of Hyderabad, DCIS, Hyderabad, 500046, AP, India
rafahmohammed@gmail.com,
{wankarcs,atulcs,crrcs}@uohyd.ernet.in

Abstract. Data Replication Broker is one of the most important components in data grid architecture as it reduces latencies related to file access and file transfers (replica). Thus it enhances performance since it avoids single site congestion by the numerous requesters. To facilitate access and transfer of the data sets, replicas of data are distributed across multiple sites. The effectiveness of a replica selection strategy in data replication broker depends on its ability to serve the requirement posed by the users' jobs or grid application. 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. In this paper we propose an approach that extends the data replication broker with policies that factor in user quality of service by reducing time costs when transferring data. The extended broker uses a replica selection strategy called Efficient Set Technique (EST) that adapts its criteria dynamically so as to best approximate application providers' and clients' requirements. A realistic model of the data grid was created to simulate and explore the performance of the proposed model. The policy displayed an effective means of improving the performance of the network traffic and is indicated by the improvement of speed and cost of transfers by brokers.

Keywords: Data Grid, Replica Selection technique, Association Rules, Broker.

1 Introduction

Grid Computing emerges from the need to integrate collection of distributed computing resources to offer performance unattainable by any single machine [1]. Data Grid technology is developed to share data across many organizations in different geographical locations. The idea of replication is to move and cache data close to users to improve data access performance and to reduce the latency of data transfers. It is a solution for many grid-based applications such as climatic data analysis and physics grid network [2]. The scientists require downloading perhaps up to 1 TB of data at a time for locally responsive navigation and manipulation. When

C. Sombattheera et al. (Eds.): MIWAI 2011, LNAI 7080, pp. 286–297, 2011.

different sites hold the required data (replicas), the selection process has a direct role on the efficiency of the service provided to the user [4]. Our experimental data has been taken from sites in world seen from *CH.CERN.N20* [7].

The rest of the paper is organized as follows: *Section 2* summarizes the statement of problem. *Section 3* explains the general aspect of the data grid architecture including the proposed approach. Our Intelligent optimizer with its two phases is explained in *Section 4, 5 and 6*. Simulation input is shown in *Section 7* and the results and their interpretation are presented in *Section 8*.

2 Problem Statements

In the context of data grid computing one of main key decision making tool in a data replication scheme is the *resource broker* that determines how and when to acquire grid services and resources for higher level components. Our proposed approach focuses on enhancing a resource broker to achieve the following:

1- Transfers of large amount of data (Terabyte or above) at a high speed.
2- Different sources may send the requested file(s) simultaneously.
3- Multiple data streams are used with each TCP connection between computing element and replica site to utilize the bandwidth.
4- Data consumers are allowed to get portions of data from different locations.
5- Dynamic and automatic replica data management.

3 Architecture of the Modified Data Replication Broker

In this section an extended version of a general Data Grid architecture is explained with functionality of its main components. The data replication broker (resource broker) resides in the middleware of data grid architecture and works as a resource manager. Figure 1 describes the services that are most commonly used in the proposed data selection scheme and shows where the data replication broker is situated. The data replication broker does the file transfer functionality in three cases:

1- Broker receives a request from data grid user/application.
2- Scheduler has signed for a replication needed somewhere in the grid sites.
3- Huge data has been generated and should be distributed in different grid sites. In all situations, the resource broker is given the identity of the file to be transferred from the “best” replica provider(s) [8][17][18][19].

3.1 Replica Management System (RMS)

As we see in the Figure 1, the main component of the Data Grid is the *Replica Management System (RMS)* [5] it acts as a logical single entry point to the system and interacts with the other components of the system. Some terms are used in Figure 1 that should be clearly defined, such as:

3.2 Replica Location Service (RLS)

RLS is the data grid 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 files are created. Later, users or services query *RLS* servers to find physical locations of replicas. It has:

- A *Logical file Name (LN)* is a unique identifier for the contents of a file (replica).
- A *Physical file Name (PN)* is the location of a replica on a storage system [8].

3.3 Replica Optimization Service (ROS)

ROS is used in the Optimization component to optimize replica selection process by minimizing different types of costs such as, *Price* and *Time* and so on. In our model, *ROS* is used to minimize the cost of the time. Minimizing the total transfer time of requested files is the main objective of our model which can be achieved by pointing the user/application requests to appropriate replica with respect to the network latency [19]. ROS gathers the information from the network monitoring service.

3.4 Network Monitoring Service (NMS)

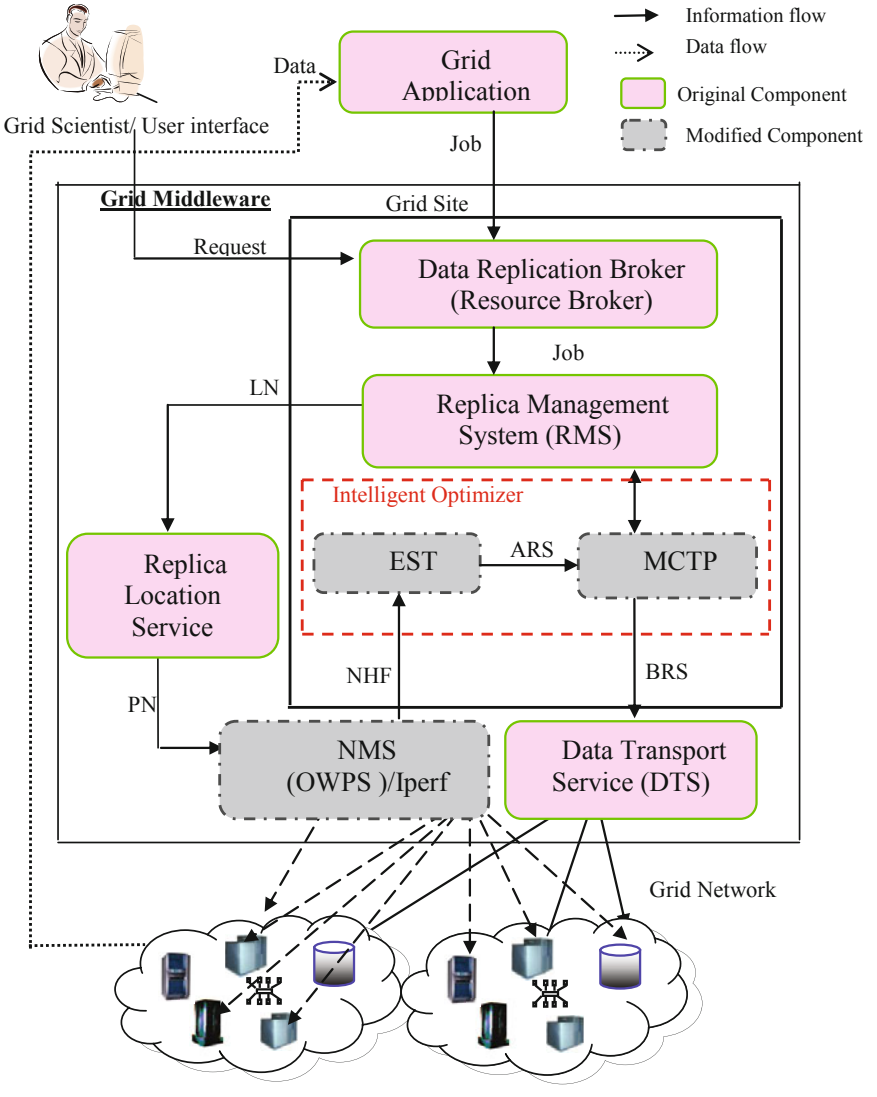
Network Monitoring Service such as*, Iperf Service* [10], is used for monitoring network conditions of connected grid nodes. The *Bandwidth (BW),* the *Distance* (Hops), the *Round Trip Time (RTT)* are examples of network conditions whose exact values can be measured using network services. In our model we propose a new network monitoring service called *One Way Ping* Service, *OWPS* to be used in *NMS* component. *OWPS* uses *One-Way Active Measurement Protocol (OWAMP)*[11]. *OWPS* is a command line client application and a policy daemon that is used to determine the one way latencies between hosts. In roundtrip-based measurements which can be measured by NWS/*Iperf* service, it is hard to isolate the direction in which congestion is experienced. One-way measurements would solve this problem and make the direction of congestion immediately apparent. This would prompt a better allocation of replicas by decreasing areas of congestion where ever possible.

3.5 Data Transport Service (DTS)

Standard tool for data transfer between the Gird sites or downloading from Grid sites is *GridFTP*. *GridFTP* is a high- performance, secure, reliable and multi streams, included in most of the storage management systems. It uses *TCP* as the transport layer protocol.

4 Intelligent Optimizer

This research shows a new optimization technique that considers link throughput (network latencies) when selecting the best replica. The new approach has two phases which are:



Where, ***LN***: Logical file Name, ***PN***: Physical file Name, ***NHF***: Network History File, ***EST***: Efficient Set of replicas Technique, ***MCTP***: Minimum Cost and Time Policy, ***ARS***: Associated Replica Sites and ***BRS***: Best set of Replica Sites.

Fig. 1. Architecture of the enhanced Data Replication Broker

1- Coarse-grain selection criteria: It is for sifting replica sites which have low latency (uncongested links).
2- Fine-grain selection criteria: It is for extracting the associated replica sites have lowest prices.

The associated sites can work together to share transferring large file to be processed by dividing the file among them and each replica site sends only a part.

5 Coarse-Grain Phase Using Efficient Replica Set Technique (EST)

The first phase of the model is called Coarse-grain selection phase. In this phase *EST* [12] selection strategy is used to extract the replica sites having good latency. Association rule concept of data mining approach is used with the following metrics:

1. *Single Trip Time (STT)*
Using *OWPS* we get *Single Trip Time (STT), STT* is time taken by the small packet to travel from *Replica Sites (RS)* to the *Computing Site (CS)*.

2. *Standardization Data*
Using a mapping function we can convert *STT/RTT* values to logical values and save the result in *Logical History File (LHF)*.

3. *Association Rules Discovery*
One of popular association rules algorithms of data mining approach is an Apriori algorithm [12]. Here, it is used for discovering associated replica sites to work concurrently and minimize total time of transferring the requested file(s).

4. *EST algorithm*
It is to extract the best set of replica sites to work concurrently and get the minimum transfer time of getting the requested files as shown in Figure 1.

6 Fine-Grain Phase Using Minimum Cost and Time Policy (MCTP)

MCTP represents the Fine-grain process. It is extracting the best replicas sets, *BRS* from associated replicas sites *ARS* is explained. The sites in *BRS* are used to send parts of requested large file or multiple related files to minimize the total transfer time. The following functions are used to represent the Fine-grain phase.

6.1 Delay Function

To determine the amount of data that can be transmitted in the network the *Bandwidth Delay Product (BDP)* should be calculated. *BDP* plays an especially important role in high-speed / high-latency networks, such as most broad band internet connections. It is one of the most important factors of tweaking *TCP* in order to tune systems to the type of network used. The *BDP* simply states that:

$$BDP = BW \times RTT \tag{1}$$

Initialize: $k=1$, $J=\{J_1,J_2,\ldots,J_x\}$, where x is max. number of jobs in a *Queue (Q)*

Step I While ($Q \neq \{\}$) OR ($k \leq x$) Do

Step II Receive $J_k=\{f_1,f_2,\ldots,f_c\}$ where c is the number of requested files in J_k

Step III Get S_i, *where* $i=\{1, 2, \ldots,M\}$ *and* M represents number of replicas.

Step IV Get *Network History File (NHF).*

- Rows = STTs / RTTs
- Columns = S_i

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

a) Calculate the Mean:

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

b) Calculate the Standard deviation:

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

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

d) Find $AVi = \frac{\sum_{j=1}^{M} Q_{i,j}}{M}$

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

Step VI Call *AT (LHF, c, s ,ARS)*

Input- LHF: Logical values of Network History File

c: Minimum confidence value.

s: Minimum support value.

Output- ARS: List of Associated Replica Sites, A_j , $j=\{1,2,\ldots,n\}$, n represents number of associated sites and $n \leq M$..

Step VII Call *Fine-grain* (A_j).

Step VIII $k=k+1$

Step IX Get next job J_k from the Q.

Fig. 2. The steps of the proposed selection algorithm

TCP Window is a buffer that determines how much data can be transferred before the server stops and waits for acknowledgements of received packets. Throughput is in essence bound by the *BDP*. Equation 1 is also used to determine the optimal size of receiver window size, *RWIN* to utilize the bandwidth [14] and *RTT* is the average round trip time of links. In case the *RWIN* is less than *BDP* that means the bandwidth is not fully used with single *TCP* stream, so multi data streams should be used [13].

6.2 Network Efficacy

The efficacy of the network can be calculated by:

$$f(s) = (k \times s / b) / (k \times s / b) + RTT \tag{2}$$

Where, ***k*** is number of bits in the sending frame. ***s*** is sending window size, ***b*** is the bit rate of the link and *RTT* is the average round trip time of links.

6.3 Multiple Data Stream Function

Multiple data streams can further utilize the bandwidth by the grid environment see Figure 3. *GridFTP* has an ability to send data using multiple streams. The number of streams, Ns, can be automatically calculated using the following formula [14]:

$$Ns = Bandwidth \times RTT \,/\, windowsize \tag{3}$$

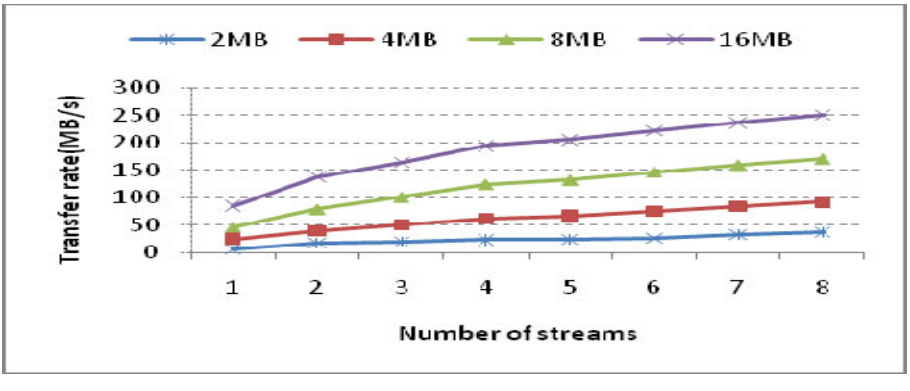


Fig. 3. Effects of varying window size and number of streams on the data transfer rate using GridFTP

7 Simulation Inputs

A *Large Hadron Collider (LHC),* operated by the European Laboratory for Nuclear Research *(CERN)* [9], generates around thirty terabytes of data which needs huge storage and large number of processors to analyses it. This data will be accessed by

researchers anywhere in the world, for downloading, reconstruction and analysis, so researchers can sit at their laptops, write small programs or macros, submit the programs through the agent, find the necessary data on servers and then run their jobs through supercomputer centers. The massive data sets are now being collected and distributed to researchers around the world through high-speed connections to the *LHC Computing Grid (LCG)*, a network of computer clusters at scientific institutions. The network employs the connectivity of private fiber-optic cable links, as well as existing portions of the public Internet.

Time and cost are the most important factors in the most of the selection processes in grids.ïIn this paper we present a dynamic replica selection strategy to enhance a replica broker. The new strategy 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. The enhanced replica broker takes into account these two factors in two phases:

1. *Coarse-grain:* It utilizes a data mining approach called the association rules to select the set of replicas from a number of sites that hold replicas.
2. *Fine-grain:* It works as a scoring function. It utilizes a grid core services such as replica location service and network monitoring services with transport service to select best replicas with respect to the cost and transfer time from uncongested set of replica sites.

To get logs files of the data grid networks and use them as an input files to our simulation, the site of *CERN* and 105 sites connected to it are used as a test bed form. All other site characteristics such as: number of jobs to be run, delays between each job submission, maximum queue size in each computing element, size and number of requested files and speed of I/O storage operations, assumed same for all replicas to see the effect of network resources only.

8 Performance Evaluation and Results

The implementation and performance evaluation of the proposed model are described in this section.

A) Experimental Data

To test the performance of our model on real data grid environment, Internet end-to-end performance monitoring reporting, *PingER* [7] is used to monitor the links between *CERN* and other sites [9]. The reports of *Feb.2011* are saved as text files. The implementation procedure for our model is done by writing a *C++* program with the following functions:

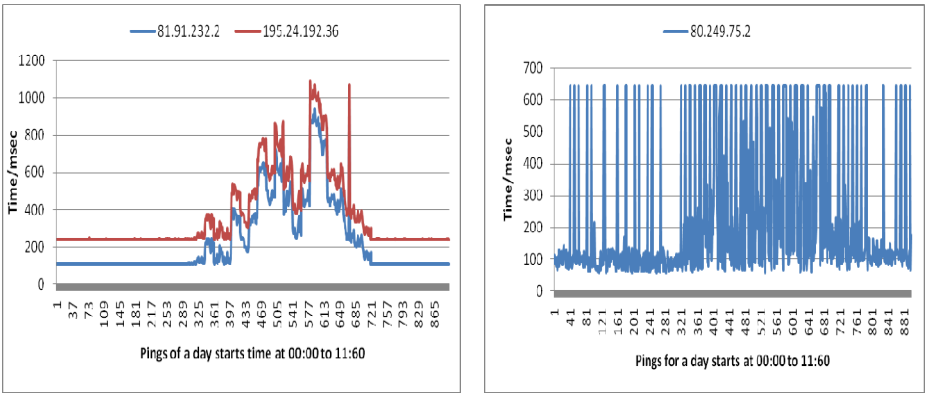
1. ***Extracting RTT function*:** This function extracts the *RTTs* from *PingER* report and save it in a text file called *Network History File (NHF).*
2. ***Converting function:*** A mathematical standardization method is used to convert the real values of *RTTs* to logical values and save it in a text file called, *Logical History File LHF*.

3. ***EST function:*** The *LHF* with minimum *confidence (c)* and *support (s)* are used as input parameters to generate Associated Replica Sites, *ARS by* executing the *EST.*

B) Network Conditions effects

After examining *PingER* reports we noted that some sites at the same time having stability in the network links. In other words, at certain time of the day, some sites have *Round Trip Times* with almost constant values (good latency) as shown in *Figures 3.* It shows the status of the link between *CERN site* and *"81.91.232.2", "waib.gouv.bj".* As it is noted the stability of the link varies from time to time, so the link between the two sites was stable in the beginning of *2Feb2011,* then it became unstable in the mid of the day and after that again became stable and also the "*195.24.192.36", "www.camnet.cm*" site,, has also a stabile link whereas the "*80.249.75.2*", "*univ-sba.dz"* site has unstable link in the same time.

When user/application request is received at the beginning of the day by our proposed broker, both of stable links sites will be selected and appear in *ARS* after applying the *Coarse-grain* phase whereas site "*univ-sba.dz*", will not be selected because it is unstable at the same time.



(a) (b)

Fig. 4. RTT between number of data grid sites and "cern.ch" site

Another network condition effect is observed between distributed sites, the transfer rate of requested data files varies with the *TCP* window size of the sender and the number of streams used to transfer data as shown in *Figure 4* above.

The simulator consists of a hundred and five sites spread around the world that deal with *cern.ch, 192.91.244.6* via internet.

We simulate our work as the following steps:

1- The input log files are

a- Latency log file: In this file all round trips time between *cern.ch 192.91.244.6* and distributed sites for the date of date (2Feb2011) are saved.

b- Replica location log file: In this file we saved the IP addresses of sites with the name of their files which can be used for others. The cost per *MB* of the file, the *Bandwidth BW*, window size and *Maximum Transmission Unit MTU* [14].

2- Output file has set of association rules

To get the output file, let us see this scenario:

A data grid job *(J_1)* contains five file of 10GB size *(f_1,f_2,f_3,f_4 and f_5) is* submitted to the *Replica broker*. The required files are distributed on the 105 replica sites (a site holding a copy of the file is called a replica). To execute J_1 on the computing element "*cern.ch*" that is referred by S_0, the required files should be available at S_0. After checking replica location log file, the files are found in the thirty distributed sites only, as shown in Figure 5. In order to minimize total executing time of J_1, the files should be concurrently taken from different sites. Selecting number of sites from these thirty sites represents first phase of our work, *Coarse-grain.*

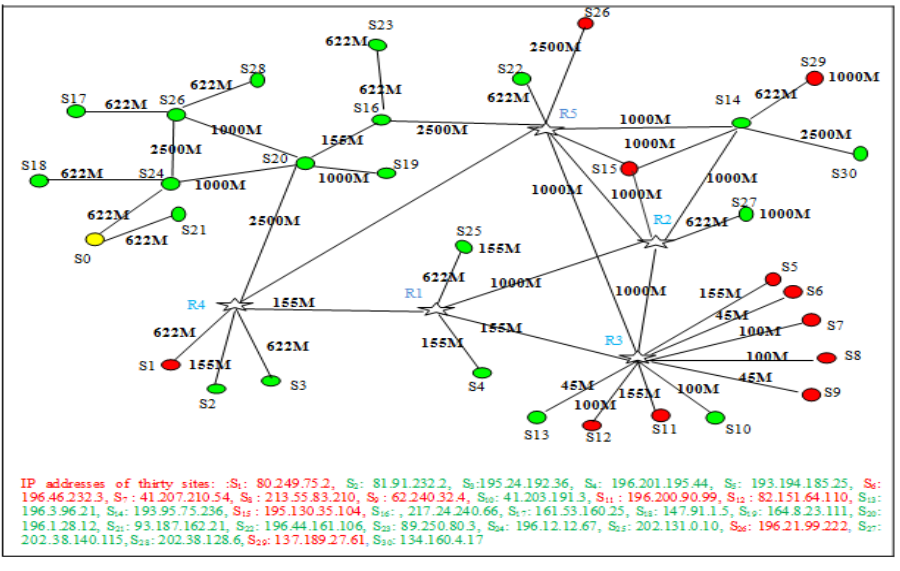


Fig. 5. The EU Data Grid Test bed of 30 sites connected to CERN and the approximate network Bandwidth

a- Simulation result of the Phase one: Coarse-grain

When J_1 arrives at *2:00 Am,* the selected set of replica sites having a good latency are:*{S_2, S_3, S_4, S_5, S_{10}, S_{13}, S_{14}, S_{16}, S_{17}, S_{18}, S_{19}, S_{20}, S_{21}, S_{22}, S_{23}, S_{24}, S_{25}, S_{27}, S_{28}, S_{30}}*, this set is called *Associated set of Replica Sites (ARS).*

b- Simulation result of the Phase one: Fine-grain

Fine-grain is purifying the selection to get the *Best set of Replica Sites (BRS),* sites with a good price and throughput. Using same scenario to get *BRS* from *ARS* we apply *MCTP* with the following steps:

1- Apply a scoring function, Equation 4 on *ARS* to determine *BRS*, Which is referred to as S_{ij}, sites with highest scores [8]. For simplicity, assumed an equal weight for both w_D and w_p that is 0.5.

$$S_{ij} = -w_D (D_{ij} / Ns) - w_p (P_j \times f) \quad (4)$$

C) Comparison with other methods

Replica broker is varying with different selection strategies that are used to get the best site having the requested file. In this section we explain the difference between our replica broker and others:

a- EST with traditional model: In traditional data grid broker who uses traditional selection method, the best replica is the one which has the least number of *Hops* (routers), or the highest Bandwidth or the minimum *Round Trip Time (RTT)* to reach the computing site [15]. Figure 5 shows 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 model the site which has the highest bandwidth does not always mean to be the best because sometimes this highest bandwidth link can be congested.

***b- EST with Random model*:** In this model the replica is selected randomly to serve user's request [16]. The drawback of this model is it does not take care of network conditions like BW and congested links. It selects random provider from the list of providers to get the requested file as shown in Figure 6.

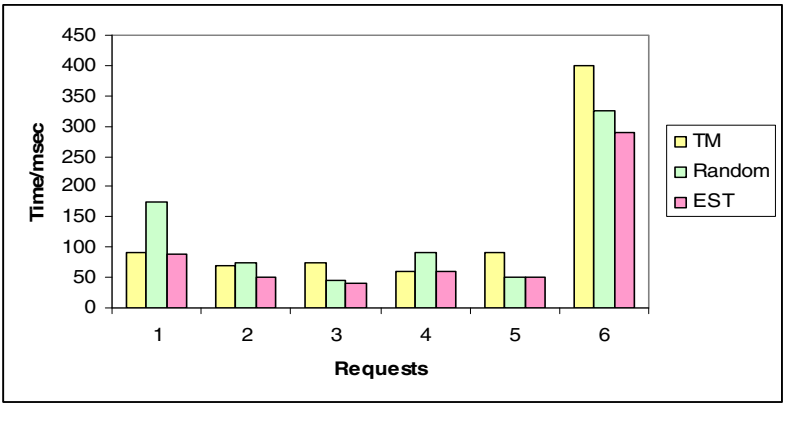


Fig. 6. Comparison of EST with TM and Random

Acknowledgments. 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. Buyya, R., Venugopal, S.: The Gridbus toolkit for service oriented grid and utility computing: an overview and status report. In: 1st IEEE International Workshop Grid Economics and Business Models, GECON 2004, pp. 19–66, 23 (April 2004)
2. Abbas, A.: Grid Computing: A Practical Guide to Technology and Applications (2006)
3. Venugopal, S., Buyya, R., Ramamohanarao, K.: A taxonomy of Data Grids for distributed data sharing, managment, and proce. ACM Comput. Surv. 38(1), Article 3 (June 2006)
4. Vazhkudai, S., Tuecke, S., Foster, I.: Replica selection in the globus data grid. In: First IEEE /ACM Int. Conf. on Cluster Computing and the Grid, CCGrid 2001 (2001)
5. Rahman, R.M., Barker, K., Alhajj, R.: Replica selection strategies in data grid. Journal of Parallel and Dis. Computing 68(12), 1561–1574 (2008)
6. Almuttairi, R.M., Wankar, R., Negi, A., Rao, C.R., Almahna, M.S.: New replica selection technique for binding replica sites. In: 2010 1st International Conference on Data Grids, Energy, Power and Control (EPC-IQ), pp. 187–194 (December 2010)
7. `https://confluence.slac.stanford.edu/display/IEPM/PingER`
8. Lin, H., Abawajy, J., Buyya, R.: Economy-Based Data Replication Broker. In: Proceedings of the 2nd IEEE Int'l Con. on E-Science and Grid Computing (E-Science 2006), Amsterdam, Netherlands. IEEE CS Press, Los Alamitos (2006)
9. Earl, A.D., Menken, H.L.: Supporting the challenge of LHC produced data with ScotGrid, The University of Edinburgh CERN-THESIS-2006-014 (April 2006)
10. Tirumala, A., Ferguson, J.: Iperf 1.2 - The TCP/UDP Bandwidth Measurement Tool (2002)
11. `http://www.internet2.edu/performance/owamp`
12. Almuttari, R.M., Wankar, R., Negi, A., Rao, C.R.: Intelligent Replica Selection Strategy for Data Grid. In: Proceeding of the 10th Int. Conf. on Parallel and Distributed Proceeding Techniques and Applications, LasVegas, USA, vol. 3, pp. 95–100 (July 2010)
13. Matsunaga, H., Isobe, T., Mashimo, T., Sakamoto, H., Ueda, I.: Data transfer over the wide area network with large round trip time. In: IOP Science, 17th Int. Conf. in High Energy and Nuclear Physics (2010)
14. Wang, J., Huang, L.: Intelligent File Transfer Protocol for Grid Environment. In: Current Trends in High Performance Computing and Its Applications, Part II, pp. 469–476 (2005), doi:10.1007/3-540-27912-1_63
15. Kavitha, R., Foster, I.: Design and evaluation of replication strategies for a high performance data grid. In: Proceedings of Computing and High Energy and Nuclear Physics (2001)
16. Ceryen, T., Kevin, M.: Performance characterization of decentralized algorithms for replica selection in distributed object systems. In: Proceedings of 5th International Workshop on Software and Performance, Palma, de Mallorca, Spain, July 11 -14, pp. 257–262 (2005)
17. Almuttari, R.M., Wankar, R., Negi, A., Rao, C.R.: Smart Replica Selection for Data Grids Using Rough Set Approximations (RSDG). In: CICN, pp. 466–471 (November 2010)
18. Almuttari, R.M., Wankar, R., Negi, A., Rao, C.R.: Rough set clustering approach to replica selection in data grids (RSCDG). In: ISDA 2010, pp. 1195–1200 (November 2010)
19. Almuttari, R.M., Wankar, R., Negi, A., Rao, C.R.: Replica Selection in Data Grids Using Preconditioning of Decision Attributes by K-means Clustering (K-RSDG). In: Information Technology for Real World Problems (VCON), pp. 18–23 (December 2010)