International Journal of Advanced Information Science and Technology (IJAIST) ISSN: 2319:2682
Vol.4, No.12, December 2015 DOI:10.15693/ijaist/2015.v4i12.42-48

Neuro fuzzy approach to evaluate the node state for replica selection on Data Grid

Dr.K.Sashi
Associate Professor/Department of Information Technology
SNR Sons College, Coimbatore, India

***Abstract*— Data grid technology promises geographically distributed scientists to access and share physically distributed resources such as computing resources, networks, storages and data collection for large scale data intensive applications. An important technique that speeds up data access in data grid systems is to replicate data in multiple locations so that it reduces data access costs and increases data availability. Replica selection, one of the key components of replica management system is a high level service that chooses a replica from among many distributed replicas. When different sites hold replicas, there are significant benefits realized when selecting the best replica. Network performance and the node state play a major role in selecting a best replica. In this paper for replica selection, node state is evaluated using Neuro Fuzzy approach and transfer of node from one state to another is predicted using grey model. The algorithm is simulated using a data grid simulator, OptorSim, developed by European Data Grid projects.**

***Index terms* - Grid Computing, Data Grid, Replica Selection, Node state, Fuzzy system**

I. INTRODUCTION

Grid Computing is one of the fastest emerging technologies within the high performance computing environment. It utilizes computing resources that are distributed in different geographical locations but are organized to provide an integrated service. A grid is a large scale resource sharing and problem solving mechanism in virtual organization [1]. Grid deployments that require access to and processing of, data are called data grids. They are optimized for data oriented operations.

Data grid technology promises geographically distributed scientists to access and share physically distributed resources such as computing resources, networks, storages and data collection for large scale data intensive applications [2, 3]. In applications such as high energy physics, life science, global climate change etc. the volume of scientific data is measured in terabytes and even in petabytes [4, 5, 6]. With the amount of increase in such scientific data and the need for resource sharing for collaborations it becomes essential to manage and share the data efficiently.

In data grid the drawback of transferring a file from one site to another in real time are bandwidth consumption and access delay. For accessibility and cost reasons, data needs to be distributed among multiple computing sites rather than hosted in a single site. An important technique that speeds up data access in data grid systems is to replicate the data in multiple locations so that it reduces data access costs. The idea of replication is to store copies in different locations so it can be easily recovered if one copy at one location is lost. Moreover, by keeping the data closer to user via replication, data access performance can be improved. Replication also improves the reliability of the system by increasing the data availability.

A. Data replication management strategies

Replication is an efficient method to achieve high network performance in distributed environments. It can be classified as static and dynamic. Dynamic replications are suited for grid environment. Dynamic replication can improve the performance by taking into consideration the changes in grid environments and creating or placing the replicas automatically where it is required.

A data replication management strategy includes four important challenges which are as follows

- **Replica creation**

Dynamic automatic creation of replica in a suitable site by data replication strategy can increase the system performance.

- **Replica placement**

Placing the replicas in the appropriate location can reduce the network bandwidth consumption and job turnaround time. Replica placement algorithms are based on the heuristics that consider both network latency and user requests to select the best candidate sites to place replicas. One of the challenges in data replication is to find the optimal placement of replicas.

- **Replica replacement**

If the site that holds the replica does not have sufficient space to store the new replica, enough space is to be created by deleting one or more files. Replica replacement strategy will be useful if there is no enough space left for all the new files. Since the storage capacity is limited, a choice is made for the files to be deleted.

- **Replica selection**

Replica selection is one of the major functions of data replication that decides which replica location is the best for grid users. Replica selection is complicated because it can involve several components, including networks, CPU and disks. This paper discusses about replica selection.

As different sites hold replicas of a particular file there is a significant benefit realized by selecting a best replica among them. By selecting the best replica, the access latency time is minimized. The best replica will be the one that optimizes the desired performance criterion such as execution time, access cost and data transmission time.

As grid is dynamic in nature, the user request, network latency, CPU load vary dynamically. Therefore the selected site to fetch replica may not be the best site for subsequent requests for the change in the network conditions. Dynamism in such grid environment can be handled using fuzzy and neural network approaches. This paper uses Adaptive nuero fuzzy inference to predict the behavior of the node state and grey model to predict the node transition state.

The remainder of the paper is organized as follows. Section 2 gives a brief introduction of previous work on replica selection. Section 3 describes the proposed algorithm and the performance evaluation is presented in section 4. Finally, the conclusion and proposed future work are discussed.

II. RELATED WORK

Recently, there has been great interest in modeling data grid environments, in which the replica selection technique is very important. Some recent studies have examined the problem of dynamic replica selection in data grids.

Husni et al. [7] proposed a replica selection system that selects the best replica location for the users running jobs in a minimum response time that can be estimated by considering new factors besides the data transfer time namely the storage access latency and the replica requests that waiting in the storage queue. But response time is not the only criteria. Other status such as reliability and node state can be considered.

Sudharsan Vazhkudai and Jennifer [8] combine end to end application throughput observations with network and disk load variations and captures whole system performance and variations in load patterns. They develop a set of regression models to derive predictions, and they achieve relatively good accuracy. However their algorithm requires large amounts of data.

In [9] the authors emerged to improve the estimation of the expected user response time, based on measurement of other network factors such as network bandwidth and server request latency. Prediction is based on historical log files which are used to decide the best replica. They have considered the network bandwidth and dynamically choose the required replica at runtime. This study adapts to dynamic changes in bandwidth.

In GRESS [10], a replica selection service is based on the Open Grid Services Architecture, which facilitates discovery and incorporation of the service by other Grid components such as Grid planners and virtual data workflow execution environments. Authors have considered the GridFTP log file only as a prediction tool in order to find the replica in a minimum response time.

Hu and Jennifer [11] suggested a lightweight prediction based learning (IBL) algorithm to allow replica selection with less required data. In this approach they pick the replica site using a simple relative metric, instead of calculating absolute values of file transfer tiles for all replicas and they pick the site based on the relative rankings of their file transfer times. IBL approach can be an efficient tool for replica selection when only limited data sources are available.

Rahman et al. [12] have considered the storage access latency with the response time. They have considered historical data information about storage latency and data transfer time as a predictor of future time, but future prediction for storage access latency is not accurate, because the grid resources such as storage are changed and upgraded all over the time.

In [13, 14] authors use parallel download to increase the end-to-end use request time. So that the required file is downloaded from all servers that house the underlying replica simultaneously. In such approaches the required file is typically partitioned into segments and each segment will be downloaded from each available server.

Jing Li [15] uses the grey system theory which is used to predict the data response time on the basis of the GM (1, 1) grey dynamic model. Markov chain is applied to achieve state transition probability matrix to predict the reliability of replicas in the form of probability. They make a valid prediction, helpful to selection decision, and are able to achieve load balance between nodes holding replicas. But predicting the response time is not the only criteria for replica selection.

In [16] the authors focus on determining an appropriate set of replicas that at least cover the data, and farthest utilize the system parallel computing capacity. According to the authors there is a trade-off between increasing parallelism and reducing redundancy as more replicas involved in computations. In order to balance the two conflict demands, they try to measure how the replica number influences the performance using Utility Theory in economics

Husni et al. [17] addressed the replica selection problem in a grid environment where the users are competing for the limited data resource. Their aim is to establish fairness among the users in the selection decision. They use Analytical Hierarchy Process to solve the optimization problem. Replica selection is based on two QoS parameters reliability and security. The system achieves better satisfaction for grid users by providing users with the best selection of the required replica, such that the reliability and security are maximized, and the response time is minimized.

Chang et al. [18] proposed a replica services based on state evaluation strategy. Best replica is selected based on replication state information. Replica state information is obtained using node state fuzzification evaluation. State life cycle is predicted using grey prediction model. Their services reduce the

International Journal of Advanced Information Science and Technology (IJAIST) ISSN: 2319:2682
Vol.4, No.12, December 2015 DOI:10.15693/ijaist/2015.v4i12.42-48

replica access latency and realize load balance. But the fuzzy rules are formed only in the initial stage. The fuzzy rules formed once will not suite all the time due to the dynamic nature of grid.

III. REPLICA SELECTION SERVICE

Replica selection one of the key components of replica management system is a high level service that chooses a replica from among many distributed replicas. When different sites hold replicas, there are significant benefits realized on selecting the best replica. Network performance and the node state play a major role in selecting a best replica. In this paper, node state is evaluated to select a best replica. In this paper, node state is evaluated using fuzzy logic approach and node transition is predicted using grey prediction model as in [18]. In Grid environment, node state is dynamic and it is evaluated while selecting a best replica. The node state once evaluated does not remain the same. Therefore, fuzzy controller is trained for better learning ability by Adaptive Neuro Fuzzy Inference System (ANFIS) developed by Jang in 1993 [19]. The replica selection process consists of three steps:
Step 1: Evaluating the node state using fuzzy system [18].
Step 2: Fuzzy rule learning procedure using ANFIS.
Step 3: Grey prediction model to predict the node transition [18].

Step-1: Evaluating the node state using fuzzy system

In this work, fuzzy sets are used to evaluate the state of a node. Seven kinds of node states considered are as follows:

Negative Large (NL), Negative Medium (NM), Negative Small (NS)

Zero (ZO), Positive Small (PS), Positive Medium (PM), Positive Large (PL)

The two inputs of adaptive fuzzy controller are Node Static Method (NSM) representing memory, CPU and bandwidth and Node Dynamic Metric (NDM) representing load used and network accessibility.

A node can have many static and dynamic metrics but only a few are considered here. Figure 1 shows the triangular shaped membership function used. The input is normalized to -1 to +1.

In general, a fuzzy control rule is a fuzzy conditional statement in which the antecedent is a condition in its application domain and the consequent is a control action for the system under control. The control rule can be expressed as follows.

Rule R_{ij} : if NDM is A_i and NSM is B_j then nstate is C_{ij}
Where *NDM*, *NSM* and *nstate* are linguistic variables and A_i, B_j and C_{ij} are linguistic values with membership function
$A_i : C_{ndm} \longrightarrow [0,1]$, $B_j : C_{nsm} \longrightarrow [0,1]$ $C_{ij} : C_{nstate} \longrightarrow [0,1]$

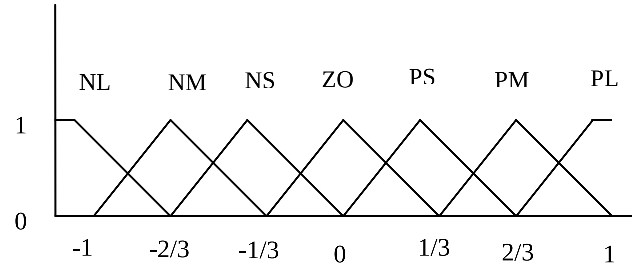

Figure 1: Tri aped fuzzy membership functions

The sample initial control rules are shown in table 1.

Table 1: Rules used in the fuzzy controller

		NDM						
		NL	NM	NS	ZO	PS	PM	PL
NSM	NL	NL	NL	NL	NL	NM	NS	ZO
	NM	NL	NM	NM	NM	NS	ZO	PS
	NS	NL	NM	NM	NS	ZO	PS	PM
	ZO	NL	NM	NS	ZO	PS	PM	PL
	PS	NM	NS	ZO	PS	PM	PL	PL
	PM	NS	ZO	PS	PM	PL	PL	PL
	PL	ZO	PS	PM	PL	PL	PL	PL

Step-2: ANFIS: Fuzzy rule learning procedure

ANFIS an Adaptive Neuro Fuzzy Inference System is used to provide the fuzzy controller with on line and learning ability. Fuzzy inference is the process of formulating the mapping from a given input to an output using fuzzy logic.

The mamdani type inference which uses expert knowledge to generate rule set and membership functions for both input and output variables is used here. It is a most commonly used approach to develop fuzzy logic models for control applications. Mamdani ANFIS architecture consists of five layers.

- The first layer is the fuzzification layer where the membership functions are defined.
- The second layer is the inference or rule layer where the firing product of the two inputs is taken.
- In the third layer which is the implication layer the consequent parameters are determined.
- The fourth layer called aggregation layer where the summation of all the iterations are calculated.
- The fifth layer is the defuzzification layer where crisp output is produced.

The input to the ANFIS is NDM and NSM. **O**utput of each layer is given below.

International Journal of Advanced Information Science and Technology (IJAIST) ISSN: 2319:2682
Vol.4, No.12, December 2015 DOI:10.15693/ijaist/2015.v4i12.42-48

Layer 1: Fuzzification layer (L_1)

Here the membership functions are defined as

$O_{1,i} = \mu_{Ai}(NDM)$ and $i=1,2...n;$

$O_{1,i} = \mu_{Bi}(NSM)$ and $i=1,2...n;$

The membership function is triangular shaped function and uses three parameters (a_i,b_i,c_i).

Where

$$\mu_{Ai}(NDM) = \begin{cases} 0 & NDM \le a_i \\ (NDM-a_i)/(b_i - a_i), & a_i \le NDM \le b_i \\ (c_i-NDM)/(c_i - b_i), & b_i \le NDM \le c_i \\ 0 & c_i \le NDM \end{cases}$$

Layer 2: Inference layer or rule layer

$$O_{2,i} : \quad \omega_i = \mu_{Ai}(NDM) * \mu_{Bi}(NSM)$$

where ω_i is the firing strength and is generated with product method.

Layer 3: Implication Layer

$$O_{3,i} = \omega_i C_i$$

The consequent parameters are determined by C_i. Implication operator is product.

Layer 4: Aggregation Layer

$$O_4 = \sum \omega_i C_i$$

Aggregate operator is sum.

Layer 5: Defuzzification Layer

$$O_5 = f = D(O_4)$$

The crisp output f is achieved with the defuzzification method, COA(CenterOfArea).

For adjusting the mamdani ANFIS model parameters, supervised learning is used. i.e. Supervised learning is used for training the rules. In supervised learning, an initial set of membership functions and rules are generated. The model is then optimized using neural network algorithms (back propagation) to minimize the error between training rules and model-generated rules.

A learning rule explains how these parameters are updated to minimize the predefined error measure. The error measure computes the discrepancy between the network's actual output and desired output. Back propagation learning is used as the basic learning rule.

The overall error measure can be minimized by

$$Ep = \sum_{k=1}^{N} (d_k - x_k)^2$$

where, N is the number of nodes in the neural network,
d_k is the *kth* component of the *pth* desired output and
x_k is the *kth* component of the predicted output.

Step-3: Grey prediction model to predict the node transition time

Grid consists of many nodes and these nodes may enter as well as leave the grid network at any time. Since grid is dynamic to predict the node state, grey model can be used. The time of the node that translates from one state to another is predicted by grey model. In grey prediction model, GM (n, m) denotes a grey model, where n is the order of the difference equation and m is the number of variables. GM (1, 1) grey model is a time series forecasting model. The steps to predict the state is calculated as in [18] and are as follows

1: Get Current time Sequence.

A small number of the passed state interval time is collected to form current time sequence, which is denoted by:

$$t^{(0)}=(t^{(0)}(1),t^{(0)}(2),t^{(0)}(3),..,t^{(0)}(n)) \quad \ldots\ldots\ldots\ldots\ldots(9)$$

Where n : the number of samples.

2: Do Accumulated Generating Operation (AGO) formation of $t^{(0)}$

Define $t^{(1)}$ *as:*

$$t^{(1)} =(t^{(1)}(1), t^{(1)}(2)\ t^{(1)}(3), \ldots\ldots\ldots\ t^{(1)}(n)) \quad \ldots\ldots\ldots(10)$$

where

$$t^{(1)}(1) = t^{(0)}(1), \text{ and } t^{(1)}(k) = \sum_{m=1}^{k} t^{(0)}(m) \text{ for } k = 2,3,\ldots.n \quad (11)$$

3: Form GM(1,1) model

From the AGO sequence $t^{(1)}$*, a GM(1,1) model can be used , which corresponds*

To the following first-order differential equation:

$$dt^{(1)}(k)/dk+at^{(1)}(k) = b \ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(12)$$

Therefore the solution of Eq.(12) can be obtained using the least square method.

i.e.

$$t_p^{(1)}(k) = (t^{(0)}(1)-b/a)*e^{-a(k-1)}+b/a \ \ldots\ldots\ldots(13)$$

where

$$[a,b]^T = (B^T B)^{-1}B^T T_n \ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (14)$$

$$B = \begin{bmatrix} -1/2(t^{(1)}(1)+t^{(1)}(2)), & 1 \\ -1/2(t^{(1)}(2)+t^{(1)}(3)), & 1 \\ . & \\ -1/2(t^{(1)}(n-1)+(t^{(1)}(n)), & 1 \end{bmatrix} \ldots\ldots (15)$$

$$T_n = [\ t^{(0)}(2), t^{(0)}(3)\ t^{(0)}(4),..,t^{(0)}(n)\ ]^T \ \ldots\ldots\ldots\ldots\ldots(16)$$

From Eq.(13) $t^{(1)}$ *can be obtained. Let t(0) be the fitted and predicted series*

$$t_p^{(0)}=(t_p^{(0)} (1), t_p^{(0)} (2), t_p^{(0)} (3),..\ t_p^{(0)} (n)) \ \ldots\ldots\ldots\ldots(17)$$

where

$$t_p^{(0)} (1) = t^{(0)}(1)$$

4 : Obtain the next node state interval time

Applying the Inverse Accumulated Generating Operation (IAGO), then

$t_p^{(0)}(k) = (t^{(0)}(1)-(b/a)) * (1-e^{a}) * e^{-a(k-1)}$ (18)

where $t_p^{(0)}(n+1)$ is the next state interval time termed as "node state useful-life"

5 : Form new prediction model.

Upon getting the $n+1^{th}$ state interval time, stores the state interval time into a sliding window (thus discarding the oldest state interval time), and form new prediction model as follows.

$t_{new}^{(0)} = \{ t^{(0)}(2), t^{(0)}(3), \ldots, t^{(0)}(n), t^{(0)}(n+1) \}$ ....(19)

Then, repeat the steps 2-4 to predict the $n+2^{th}$ state interval and so on.

Replica Selection Algorithm

1. *Submit jobs to grid*
2. *Data are produced in Master site.*
3. *Files replicated to region header*
4. *Region header maintains site and replica details*
5. *Jobs are scheduled to grid sites*
6. *Replica optimizer is activated*

   i. Replica Optimizer will evaluate the node state by using fuzzy system.

   ii. Fuzzy makes use of ANFIS for training the fuzzy rules.

   iii. if (Requested File stored locally execute the job)

   else node i gets the information about the node state to predict the node transfer state and selects the best replica
7. *Execute all the jobs*

IV. Performance Evaluation

A. Simulation Tool

To simulate the real grid environment, there are different types of simulators such as Optorsim, GridSim, GlomoSim, Network Simulator-2(NS2) are available. Among these simulators, OptorSim[20-24] developed by the European Data Grid projects is used to evaluate the performance of this research work. One of the European data grid project is the Compact Muon Soleniod (CMS)[25] experiment that is used to collect data at the Large Hadron Collider (LHC) being built at CERN (European Organization for Nuclear Research). The challenge of the CMS computing infrastructure is to cope with the very large data access requirement. The CMS run on grid testbed implementation and this testbed is used in this work.

The simulation is controlled by using four configuration files namely, Parameter Configuration, Grid Configuration, Job Configuration and Bandwidth Configuration File. The basic Simulation parameters are set up by means of parameter configuration file. The most important parameter includes number of jobs, access pattern for job, initial file distribution job delay and the job submission pattern by which the user submit jobs to the resource broker.

Grid configuration file specifies the network topology. The topology used here is the CMS testbed architecture. In this there are twenty nodes (sites) in which two of them have only storage element and which acts as master node. CERN is considered as master site where data is produced initially. Jobs are processed in the remaining sites which have both storage and computing element. There are eight routers which is used to forward request to other sites. The topology of CMS testbed is depicted in figure 2. The topology is grouped into two regions and each region has many sites. Master site has the most capacity in order to hold all the master files at the beginning of the simulation. The storage size of master site is 100 GB and all other site is 50 GB. There are 100 jobs with six different job types. Each data file to be accessed is 1 GB.

Job configuration files contain information for the simulated jobs. It also specifies the job each site will accept, the probability each job runs and the files needed by each job. There are six job types, with no overlap between the set of files each job accessed.

Bandwidth configuration file is used to describe the background network traffic. It is a site by site matrix which gives, for each pair of sites, the name of the data file containing the relevant bandwidth information and also the time difference between the reference time zone and the source site. These configuration files are read at the start of the simulation. In data grid environment, various job execution scenarios are present. The job execution scenario used for this algorithm is shown in table 2.

The order in which a job request files is determined by the job's access pattern. Various access pattern generators of OptorSim are Sequential Access Generator, Random Access Generator, Random Walk Unitary Access Generator and Random Zipf Access Pattern Generator. In the proposed work Random Zipf Access Pattern Generator is considered.

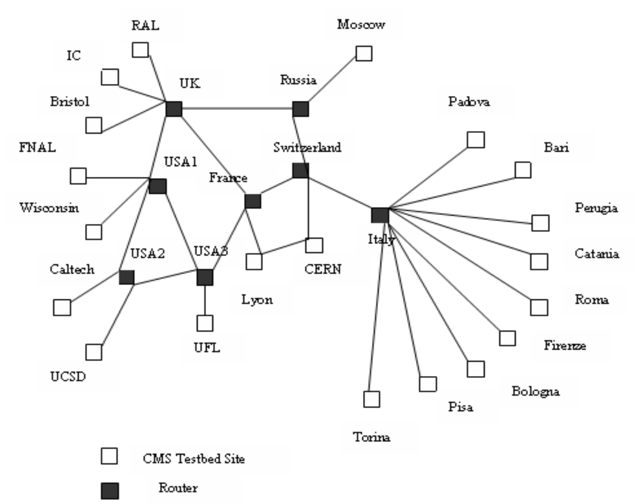

Figure 2: CMS Topology

Table 2: General Configuration of Parameters

Parameters	Values
Number of Jobs	100
Number of Job Types	6
Number of file accessed per job	10
Size of single file	1 GB
Total Size of files	1000 GB

B. Simulation Results and Discussions

Optorsim and JAMAL(Java Matlab) is used to test the performance of the Replica Management scheme and replica selection. Jamal is an open source, Java RMI based tool that makes it possible to call Matlab functions from java programs. Matlab is a commercial high level technical computing language containing a lot of libraries. It was developed by Mathworks. Matlab allows matrix manipulations, plotting of functions and data, implementations of algorithms, creation of user interfaces and interfacing with programs written in other languages such as C++, C, Java etc. Matlab functions can be called from java, passing and returning java primitives and their arrays. Since Jamal is based on the RMI technology it allows for calling Matlab functions on the fly, without saving results to a temporary file. ANFIS algorithm is implemented in Matlab and the results are passed to OptorSim for further execution.

Replica selection is a high level service that helps grid applications to choose a replica based on system performance and data access features. The optimization algorithm determines the replica that should be accessed from a given location. Replica selection makes use of transport service to transfer the files. In grid, nodes are dynamic and there is a need to evaluate the state of the node to make the replica selection effective. In this framework fuzzy system is used to evaluate the node state. The fuzzy rules formed once will not suite all the time due to the dynamic nature of grid. So, there is a need for learning algorithm to train the fuzzy rules. Adaptive Neuro Fuzzy Inference System (ANFIS) is used to train the fuzzy rules. Finally it is better to predict the state transition for the node. It is predicted using Grey model.

Job is submitted to the grid. Each job requests file of different sizes. It is assumed that each job is submitted at 25 milliseconds interval. Time taken for completing a job is equivalent to the waiting time in the queue and execution time. The systems performance is evaluated in three different scenarios such as small, medium and large. Since the file size and the number of jobs influence the data transfer time the system is analyzed by varying the file size and number of jobs each time. For each scenario the number of files for each job with 6 different job types is 500, 1000 and 2000 and the file sizes for different scenarios are:

Figure 3 shows the state report interval time. It shows the relationship between the state report interval time and the replica request time. The state report interval time does not change based on the request frequency. Figure 4 shows the replica access latency of a single file. It shows the access latency for replica requested for 15 times. When compared to LRU and Replica State Evaluation Strategy (RSES) [18], the proposed method shows better performance.

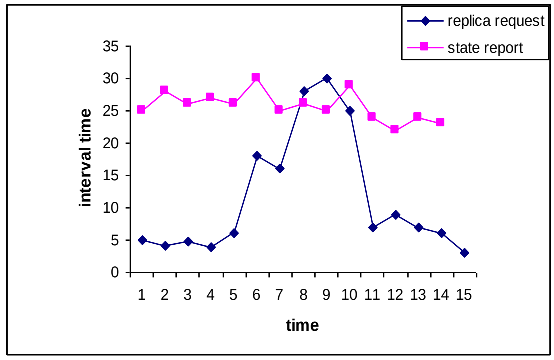

Figure 3: State report interval time

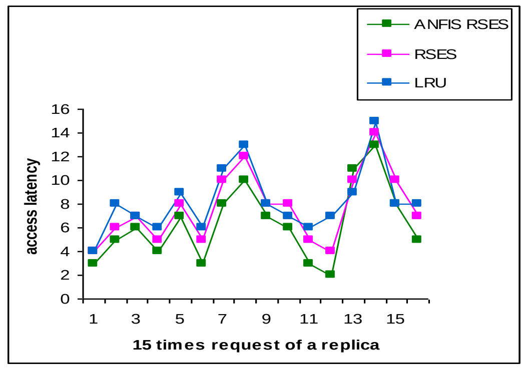

Figure 4: Replica access latency

V. CONCLUSION AND FUTURE WORK

Replica selection plays an important role in replica management. To make an effective selection of replica from different replicas, node states are evaluated. In grid, nodes are dynamic i.e. a node can enter and leave the grid at any time. Hence, node states are evaluated to make the replica selection effective. In this framework fuzzy system is used to evaluate the node state. The fuzzy rules formed once will not suite all the time due to the dynamic nature of grid. Adaptive Neuro Fuzzy Inference System (ANFIS) is used to train the fuzzy rules. Finally node state transition is predicted using Grey model. By using this approach replica access latency can be minimized which in turn reduces the job execution time. As a future work the proposed replica selection services can be implemented in real grid system.

REFERENCES

[1] I. Foster, C. Kesselman, S. Tuecke (2001) The Anatomy of the Grid: Enabling Scalable Virtual Organizations, International Journal of High Performance Computing Applications, 15(3), pp. 200-222.

[2] B. Allcock, J. Bester, J. Bresnahan, A. L. Chervenak, I. Foster, C. Kesselman, S. Meder, V. Nefedova, D. Quesnal, and S. Tuecke (2002) Data management and transfer in high performance computational grid environments, Parallel Computing Journal, 28, pp. 749-771.

[3] A. Chervenak, E. Deelman, I. Foster, L. Guy, W. Hoschek, A. Iamnitchi, C. Kesselman, P. Kunst, M. Ripeanu, B. Schwartzkopf, H. Stockinger, K. Stockinger, and B. Tierney (2002) Giggle: A framework for constructing scalable replica location services, In *Proceedings of Supercomputing (SC2002)*, pp.1-17.

[4] W. Allcock, J. Bester, J. Bresnahan, A. Chervenak, I. Foster, C. Kesselman, S. Meder, V. Nefedova, D. Quesnel, and S. Tuecke (2001) Secure, efficient data transport and replica management for high-performance data-intensive computing, In IEEE Mass Storage Conference.

[5] W. Allcock, I. Foster, V. Nefedova, A. Chervenak, E. Deelman, C. Kesselman, J. Lee, A. Sim, A. Shoshani, B. Drach, and D. Williams (2001) High-performance remote access to climate simulation data: A challenge problem for Data Grid technologies, In Proc. of SC.

[6] A. Chervenak, I. Foster, C. Kesselman, C. Salisbury, and S. Tuecke (2000) The data grid: Towards an architecture for the distributed management and analysis of large scientific datasets, Journal of Network and Computer Applications, Academic Press, New York, USA, 23(3), pp. 187–200.

[7] Husni Hamad E. AL-Mistarihi and Chan Huah Yong (2008) Response Time Optimization for Replica Selection Service in Data Grids,Journal of Computer Science, 4(6) pp. 487-493.

[8] Sudharsan Vazhkudai and Jennifer M.Scholf (2003) Using Regression Techniques to Predict Large Data Transfers, International Journal of High Performance Computing Applications - Special Issue on Grid Computing: Infrastructure and Applications, 17(3), pp. 249-268.

[9] Tim Ho and David Abramson (2005) The griddles data replication service. Proceedings of the 1st International Conference on E-Science and Grid Computing, pp. 271-278.

[10] Yong Zhao and Yu Hu (2003) Gress: A Grid Replica Selection Service. In 16th international conference on Parallel and Distributed Computing Systems, pp. 1-6.

[11] Y.Hu, Jennifer M (2004) IBL for replica selection in data intensive grid applications, Tech. Rep. TR-2004-03, The University of Chicago.

[12]Rahman R. M., K. Barker and R. Alhajj (2005) Replica selection in grid environment: A data-mining approach, Proceedings of the ACM Symposium on Applied Computing, ACM New York, USA, pp: 695-700.

[13] Zhou X. L., E. Kim, J. W. Kim and H. Y. Yeom (2006) ReCon: A fast and reliable replica retrieval service for the data grid, Proceedings of the 6th IEEE International Symposium on Cluster Computing and the Grid, May 16-19, IEEE Computer Society Press, Singapore, pp. 446-453.

[14] Jun Feng, Marty Humphrey (2004) Eliminating replica selection- using multiple replicas to accelerate data transfer on grids, Proceedings of the 10th International Conference on Parallel and Distributed Systems, July 7-9, IEEE Xplore press, USA, pp. 356-366.

[15] Jing Li (2007) A Replica selection approach based on prediction in data grid, Third International conference on Semantics, Knowledge and Grid, pp. 274-277.

[16] Dafei Yin, Bin Chen, Yu Fang (2009) A fast Replica Selection Algorithm for Data Grid, 31st Annual International Computer Software and Applications Conference, pp. 383-387.

[17] Husni Hamad E, Al-Mistarihi and Chan Huah Yong (2009) On Fairness, Optimizing Replica Selection in Data Grids, IEEE Transactions on Parallel and Distributed Computing, 20(8), pp. 1102-1111.

[14] Chang ze Wu, Kai gui Wu, Ming Chen, Chun xiao YE (2009) Dynamic Replica selection services based on state evaluation strategy, Proceedings of the Fourth Annual Conference, IEEE Computer Society, pp. 116-119.

[19] J.S.R.Jang (1993) Adaptive-Network-Based Fuzzy Inference Systems, IEEE Transaction on Systems, Man and Cybernetics, 23(3), pp. 665-685.

[20] D. G. Cameron, R. C. Schiaffino, J. Ferguson, P. Millar, C. Nicholson, K. Stockinger, and F. Zini (2004) OptorSim v2.0 Installation and User Guide, http://cern.ch/edg-wp2/optimization/optorsim.htm.

[21] The European DataGrid Project. http://www.edg.org.

[22] W. Hoschek, F. J. Jaen-Martinez, A. Samar, H. Stockinger, K. Stockinger (2000) Data management in an international Data Grid project, *Proceedings of the First IEEE/ACM International Workshop on Grid Computing(GRID '00)*, Lecture Notes in Computer Science, vol. 1971, pages 77-90.

[23] K.Holtman (2001) CMS Data Grid system overview and requirement, Tech report CERN.

[24] K.Sashi and Dr. Antony Selvadoss Thanamani (2011) Dynamic Replication in a Data Grid Using a Modified BHR Region Based Algorithm, Science Direct Elsevier –Future Generation Computer System, Volume 27(2), pp.202-210.

[25] CMS Data Challenge 2004: http://www.uscms.org/s&c/dc04/

Author Profile

Dr. K. Sashi received Ph.D in Computer Science from Mother Teresa University, Kodaikanal, India, in 2014. Her research interest includes Grid Computing , Cloud Computing and Big Data Analytics. She has published many papers in international journals.