Research Journal of Applied Sciences, Engineering and Technology 7(14): 2921-2928, 2014
ISSN: 2040-7459; e-ISSN: 2040-7467

Submitted: June 11, 2012 Accepted: July 01, 2013 Published: April 12, 2014

Trusted Virtual Machine Allocation in Cloud Computing IaaS Service

A. Radhakrishnan and V. Kavitha
Department of CSE, Anna University Chennai, Tirunelveli Region, India

Abstract: Cloud computing is a new era in computing paradigm. It helps Information Technology (IT) companies to cut the cost by outsourcing data and computation on-demand. Cloud computing provides different kind of services which includes Hardware as a Service, Software as a Service (SaaS), Infrastructure as a Service (IaaS) etc. Despite these potential benefits, many IT companies are reluctant to do cloud business due to outstanding trust issues. Cloud consumer and provider are the most interested parties to maximize their benefits. In IaaS, the cloud provider operates the whole computing platform as a resource for the customer, which is accessed by customer as a Virtual Machine (VM) via the internet. The cloud provider must predict the best machine among the available machines to launch VM. This strategic prediction would avoid exodus of computation in middle due to machine heavy load or any failure which severely affect the benefits of both consumer and provider. Since VM allocation for IaaS request is a challenging task, in this study novel architecture is proposed for IaaS cloud computing environment in which VM allocation is done through genetically weight optimized neural network. In this scenario the host load of each machine is taken as its resource information. The neural network predicts the host load of each machine in near future based on the recent past host load. It would help the VM allocator to choose the right machine. Analysis is done on the performance of genetically weight optimized Back Propagation Neural Network (BPNN), Elman Neural Network (ELNN) and Jordan Neural Network (JNN) for prediction accuracy.

Keywords: Back propagation neural network, Elman neural network, genetic algorithm, infrastructure as a service, Jordan neural network, virtual machine

INTRODUCTION

Cloud computing services have attracted the attention of IT companies due to the unique ability of providing various resources as a service. Cloud computing is derived from the service-centric perspective. From this, all resources of a cloud are provided to user as a service, to be accessed through the internet without any knowledge or control over the underlying technological infrastructure which supports them. Cloud computing provides on demand service provision in Platform as a Service (PaaS), Hardware a Service (HaaS), IaaS and so on. Even cloud computing greatly reduce the infrastructure setup cost of IT companies, they are stay away from this business owing to trust issues.

Cloud provider and cloud consumer are self interested parties to maximize their profits. In IaaS, the cloud consumer needs machine instances as a resource. This instance essentially behaves like dedicated system that is controlled by cloud consumer, who therefore has full responsibility for their operation. At this juncture, the expectation of cloud consumer is to get best resource among all available resources from cloud hub, which ultimately reduces the usage time and cost of access. In IasS the cloud provider responsibility is to select best machine to launch VM for IaaS request which reduce migration outlay due to grave load or failure of system in the middle of computation.

Predicting the future availability of resources can be extremely useful for many purposes. First, resource volatility can have a negative impact on applications executing on those resources. If the resource on which an application is executing becomes unavailable, the application will have to restart from the beginning of its execution on another resource. This wastes the valuable cycles and increases application span. Prediction can allow the scheduler to choose resources that are least likely to become unavailable and avoid these application restarts. This can increase the efficiency of the system. This research study focuses on the utilization of neural network and genetic algorithm in IaaS service VM creation. An availability predictor is framed by neural network with genetic algorithm, which forecasts the availability of resources for the period of time that an application's likely to run on them, can help schedulers make better decisions. Clearly, selecting resources that will remain available for the duration of application runtimes would improve the performance of both individual applications and of the cloud as a whole. Resource prediction is based on the resource monitoring. Resource monitoring will provide the historical data

Corresponding Author: A. Radhakrishnan, Department of CSE, Anna University, Chennai, Tirunelveli Region, India

which describes about the resources past experience. Each computing system consists of a queue data structure to keep track of past experience as load in the past N seconds. This is key parameter for system future availability prediction.

This study presents a novel architecture for IaaS request handling. In which VM creation on a machine for IaaS is based on machine future availability prediction. The prediction is done by genetically weight optimized neural networks. Analysis was also done for accuracy of prediction as wells error on prediction of various neural networks. The experiment results show that genetically weight optimized Jordon Neural Network provides better prediction in terms of error and time. This strategic way definitely enriches trust on IaaS cloud service.

LITERATURE REVIEW

Liang *et al*. (2012) proposed the design and implementation of grid resource monitoring and prediction. In this approach, Radial Basis Function (RBF) and BPNN neural network is proposed for the prediction of Grid resources. To measure the efficiency and accuracy of prediction system CPU training time and Mean Absolute Error (MAE) is used. Duy *et al*. (2009) proposed Artificial Neural Network (ANN) prediction approach for the accuracy prediction of host load in grid computing environment, Back propagation neural network is used for the prediction. Doulamis *et al*. (2004) proposed neural network with constructive algorithm for the prediction of workload in 3-D rendering of Images. In which constructive neural network algorithm with the assumption of hidden neurons is used and also proposed the methodology of dividing the data set. Dinda (2001) proposed, Auto Regressive (AR), Auto Regressive Integrated Moving Average (ARIMA) models for prediction of host load and running time of a job in a grid computing environment based on the history of data. An idea of Active database and centralized history maintenance is proposed by Bohlouli (2008). Centralized history maintenance, maintains the resources used by the job and active database stores the attributes of resources. Model for the prediction of traffic using bandwidth data by neural network was proposed by Alaknantha (2005). In this approach Multilayer Perceptron neural network is used, with the ability to extract patterns and detect available bandwidth. Ming (2006), described the mean based method for the prediction of resource availability, in which four system parameters such as local job arrival rate, machine utilization, standard deviation of service time, the computing capacity of machine are taken for prediction of resources. Che (2010) proposed Multi step ahead prediction of resource in the grid environment, the historical monitoring data are preprocessed to use in the Support Vector Machine (SVM), where the structural risk minimization based on Nu-support Vector Regression is proposed, which controls the complexity and upper bound data for processing. Jin (2007) proposed scheduling decision method that utilizes the Nonlinear Auto Regressive Exogenous (NARX) neural network based load prediction. NARX neural network based predictor learns the model of the system from the external input data, which resembles feed forward structure and captures the dynamics features of load. Neural Network with dynamic memory is proposed by Jeffrey (1990) where hidden units are feedback themselves which leads to the quick convergence of learning. Ioan (2004) proposed Optimization by Genetic algorithm for feed forward network to optimize the number of neurons in the layer as well as number of layers with error as fitness function. Zhen-Guo (2011) provides genetic weight optimization for feed forward network which is better than BPNN learning.

MATERIALS AND METHODS

Cloud architecture: In our architecture, depicted in Fig. 1, consist of various components. The functionalities of each component are well defined.

Cloud components functionality:
Request receiving and authentication unit: Receiving request from user who are in need of IaaS service from cloud provider. The authenticity of the user is examined based on cloud policy either third party or internal authentication. If the user identity is genuine, the request is forwarded to resource availability checking unit.

Resource availability checking unit: Keeps track of the availability of resources, here the authenticated users can submit their requested infrastructure. If the hub able to provide expected infrastructure then it triggers resource selection unit, otherwise request is redirected to another cloud hub. Resource availability database of this unit is updated by resource allocation and de allocation unit.

Resource selection unit: This unit activates resource load prediction process which is deployed in all resources in cloud hub. This unit pass N as a parameter to all prediction process which indicates how long past host load have taken for prediction process. After prediction result obtained from all resources, this unit selects one resource based on less load as well as cloud policies. The selected resource identity will send to resource allocation unit for create VM.

Resource load prediction process: This process exits in all computing resources. The main objective of this process is to predict the host load of resources for near

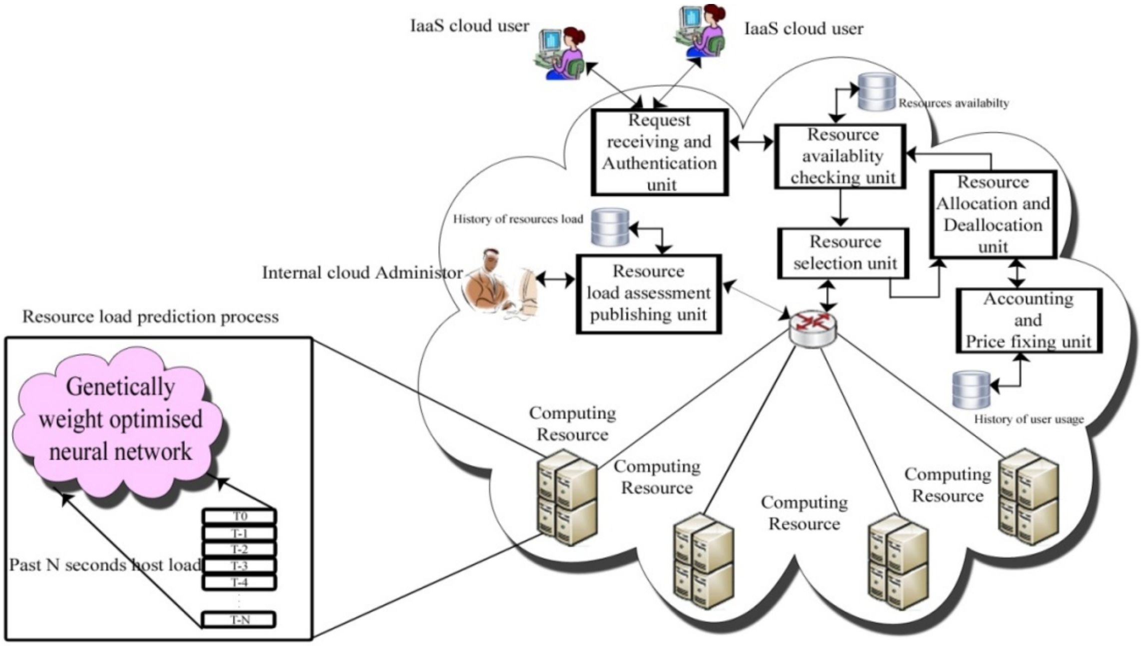
IaaS cloud user
IaaS cloud user
Request receiving and Authentication unit
Resource availablity checking unit
Resources availabilty
Resource Allocation and Deallocation unit
History of resources load
Resource selection unit
Internal cloud Administor
Resource load assessment publishing unit
Accounting and Price fixing unit
Resource load prediction process
Genetically weight optimised neural network
History of user usage
Computing Resource
Computing Resource
Computing Resource
Computing Resource
Computing Resource
T0
T-1
T-2
T-3
T-4
T-N
Past N seconds host load

Fig. 1: IaaS cloud computing provider architecture

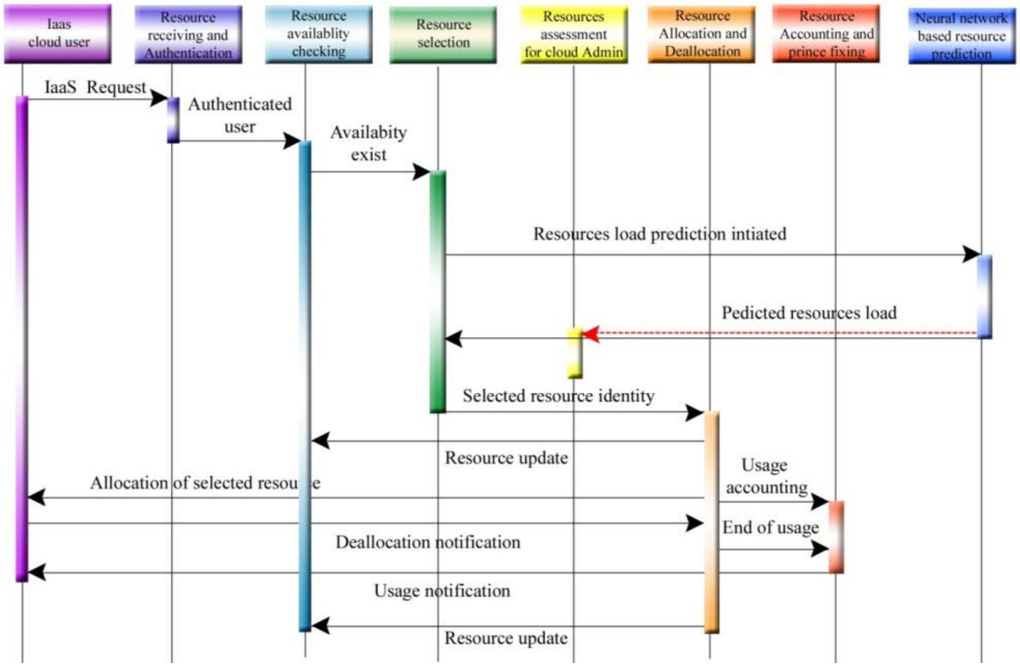
Iaas cloud user
Resource receiving and Authentication
Resource availablity checking
Resource selection
Resources assessment for cloud Admin
Resource Allocation and Deallocation
Resource Accounting and prince fixing
Neural network based resource prediction
IaaS Request
Authenticated user
Availabity exist
Resources load prediction intiated
Pedicted resources load
Selected resource identity
Resource update
Allocation of selected resource
Usage accounting
End of usage
Deallocation notification
Usage notification
Resource update

Fig. 2: Sequence diagram of resource allocation

future based on past experience of host load at N seconds. Genetically weight optimized neural network is used for this process.

Resource allocation and reallocation unit: Create and Allocate a VM to aspirant user based on the resource identity obtained from resource selection unit. This activity reflects in accounting and resource availability checking unit for metering and update respectively. This unit take care reallocating resources after usage is over.

Resource load assessment publishing unit: This unit publish the predicted load of each resources to internal cloud administrator whenever resource prediction process is initiated. This helps the internal cloud administrator to take effective decision for resource management.

Accounting and price fixing unit: Accounting information such as how long the resource was used is maintained by this unit. The cost for usage also calculated based on cloud provider policies, which will be informed to cloud user after usage.

System workflow: The work flow of the proposed architecture during resource allocation process is presented in this section. The sequence diagram in Fig. 2 depicts entire work flow. The detailed description is as below:

- The IaaS cloud user logs on cloud hub by providing valid authentication information. Authentication is valid then the user can submit their request, availability checking request created accordingly and passed to availability checking unit
- Availability checking unit examine availability of resources as per the request of user. If the availability exist then it generate request to resource selection unit, otherwise request redirected to another cloud hub
- Resource selection unit initiate resource prediction process deployed in all resources. It fixes the value of N and pass as parameter to all prediction process, which indicates how long past N second load used for near future prediction
- Prediction process is activated for all resources, which send past N second load from local queue storage reside in each resources to genetically weight optimized neural network. This network predicts a load of resource at near future based on previous experience and sends it to resource selection unit as well as resource assessment publishing unit
- Resource assessment publishing unit acquire the load of all resources and alert internal cloud administrator. These details are stored in local database for future reference
- Resource selection unit obtained near future load of all resources and choose a resource for allocation based less load and cloud selection policies. The selected resource identity sends to resource allocation unit
- Resource allocation unit allocate a selected resource to user. This activity notification sends to resource availability checking unit and accounting unit
- Resource availability checking unit update the resource availability with respect to newly allocated resource
- Accounting unit metering according to usage of resource
- The IaaS user sends notification for end of resource usage to de allocation unit
- Resource de-allocation unit release the resource from user, this activity notified to resource availability unit and accounting unit
- Accounting unit calculate cost for usage as per polices of cloud provider and send it to cloud user for payment

Prediction using feed forward and recurrent neural networks: Artificial Neural Network (ANN) is a system that approximates the operation of the human brain. ANN can accommodate much input in parallel and encode this information in distributed fashion; the information that is stored in ANN is shared by many of its processing unit. The aim of ANN is to learn from events that have happened in past and to able to apply this to future situation. ANN consists of three set of nodes namely input, hidden and output. Input and output nodes are made for receive and produce input and output respectively. The hidden nodes (called neurons) are playing vital role in ANN, which process input information.

Feed forward networks: There are neural networks wherein for every input vector laid on the network, an output vector is calculated and this can be read from the output neurons. There is no feedback; a forward flow of information is present. Networks having this structure are called as feed forward networks. BBNN is comes under feed forward network category.

Back propagation neural network: Back Propagation was created by generalizing the Widrow-Hoff learning rule to multiple layer neural networks and nonlinear differentiable transfer functions. Input vectors and the corresponding target vectors are used to train a network until the associated input vectors getting closer with specific output vectors. Neural networks with a sigmoid layer and a linear output layer are capable of approximating any function with a finite number of discontinuities. Inputs are applied to the input layer. The hidden node (Z_j) sums its weighted signals Z_{-inj} and applies the Sigmoidal activation function, as shown in the following equations:

$$Z_{-inj} = V_{oj} + \sum_{i=1}^{n} X_j V_{ij} \tag{1}$$

$$Z_j = f(Z_{-inj}) \tag{2}$$

where,

V_{0j} : Bias on hidden unit
X_j : Input unit j
V_{ij} : Weight of input unit j
$\sum_{i=1}^{n} X_J V_{ij}$: Sum of weighted input signal to hidden f $(z_{-inj}) = \frac{1}{1+e_{-}^{-z_{-inj}}}$

The output of the hidden unit is send to the output node. The output nodes (y_k) sums its weighted input signals (y_{-ink}) and applies linear activation function to calculate the output signal (y_k), as shown in the following equations:

$$y_{-inj} = w_{ok} + \sum_{j=1}^{p} Z_j w_{jk} \quad (3)$$

$$y_k = f(y_{-ink}) \quad (4)$$

where,

w_{ok} : Bias on output unit k
z_j : Hidden unit j
w_{jk} : Weight of hidden unit k
$\sum_{j=1}^{p} z_j \; w_{jk}$: Sum of weighted input signal to output f (y_{-ink}) - $\frac{1}{1+e^{-y_{-ink}}}$

Standard back propagation algorithm is used for the training of back propagation network.

Recurrent neural network: Recurrent neural networks (sometimes are these networks called feedback neural networks) can be distinguished from feed-forward neural networks in that they have a loopback connection. In its most general form recurrent network consist of a set of processing units, while the output of each unit is fed as input to all other units including the same unit.

Elman neural network: One of the most known recurrent neural networks is Elman neural network. Typical Elman network has one hidden layer with delayed feedback. At a specific time t, the previous activations of the hidden units (at time t-1) and the current input are used as inputs to the network. At this stage, the network acts as a feed forward network and propagates these inputs to produce the output. After training the activations of the hidden units at time *k* are sent back through the recurrent links to the context units and saved there for the next training (at time step t+l) (Neto *et al*., 2004).

Jordon neural network: Jordon network is Elman kind of recurrent network having feedback from output value. So at a specific time t, the previous output value and the current input are used as inputs to the hidden node. After obtaining the output for a given set of inputs, the output value is sent back through the recurrent links to the context units and saved there for the next training (at time step t+l). It differs from Elman network in knowledge accumulation at context layer, that is instead of receiving information from hidden layer it receives information from output layer.

Weight optimization of neural network using genetic algorithm:

Genetic algorithm: Genetic Algorithm (GA) was introduced by John H. Holland in 1960's where GA was a probabilistic optimization algorithm. GA is a family of computational model inspired by evolution. The original idea came from biological evolution process in chromosomes. GA exploits idea of the survival of fittest where best solutions are recombined with each other to form new better solutions. There are three processes in GA which are selection, crossover and mutation. In the standard GA, the population is a set of individual number. Each individual represents the chromosome of a life form. There is a function that determines how fit each individual and another function that selects individuals from the population to reproduce. The two selected chromosomes crossover and split again and next the two new individuals mutate. The process is then repeated until the stop condition is met. There are several terms in GA. Fitness is a measure of the goodness of a chromosome where it measure how well the chromosome fits the search space or solves the problem. Selection is a process for choosing a pair of organisms to reproduce while crossover is a process of exchanging the genes between the two individuals that are reproducing. Mutation is the process of randomly altering the chromosomes

Weight optimization of neural network: GA also has been used to produce best NN architecture and for NN weight optimization. GA starts at multiple random points (initial population) when searching for a solution. Each solution is then evaluated based on the objective function. Once this has been done, solutions are then selected for the second generation based on how well they perform. After the second generation is drawn, they are randomly paired and the crossover operation is performed. This operation keeps all the weights that were included in the previous generation but allows for them to be rearranged. This way, if the weights are good, they still exist in the population. The next operation is mutation, which can randomly replace any one of the weights in the population in order for a solution to escape local minima. Once this is complete, the generation is ready for evaluation and the process continues until the best solution is found.

Procedure for weight optimization in neural network: The genetic algorithm procedure for the weight optimization is as follows:

Step 1: Start with a population of randomly generates initial population with a number of Chromosomes (random weight for NN).
Step 2: Define fitness function based on the error on the training of NN.
Step 3: Calculate the fitness of each individual chromosome.
Step 4: Based on fitness, select a pair of fit chromosomes for combination process.
Step 5: Perform crossover and mutation to generate new chromosomes.
Step 6: Place the new chromosomes in the new population (the next generation).
Step 7: Repeat step 4 until size of new population equal to size in initial population.
Step 8: Replace initial chromosome (parent) with new populations.
Step 9: Go to step 3 until the termination condition met, terminal condition will be the minimum value of error.

RESULTS AND DISCUSSION

Experimental setup: Considering that cloud computing environment for IaaS service consist of loosely distributed systems, host load of a computing system is the most reliable information for assess the system. For host load data set, we choose “mystere 10000.dat” (A Report: Host Load Data Set, Load Dataset, http://people.cs. uchicago.edu/~lyang/load/, Feb.2009), a trace of workstation node. Workstation is a most typical computing node.

The 200 samples of load were sequentially taken from “mystere10000.data” to form experimental dataset. The transformation of selected dataset for neural network training is specified in Table 1. Experiment was running on a single IntelCore I5, 2.67 GHZ processor under windows 7 operating system. All the neural network and genetic algorithm are coded and executed in MATLAB R2010 b.

The Feed Forward and Recurrent Neural Network training, five input nodes, five hidden nodes and one output node are taken. The learning rate (α) of 0.2 is taken. The training CPU time is used to measure efficiency and Mean Absolute Error (MAE) used to measure accuracy. To measure how successful fitting is achieved between Target and Prediction, the R-square statistic measurement is employed. A value closer to 1 indicates a better fit. For weight optimization of NN by GA, two site crossovers are used and the chromosomes with least error value are taken for the Genetic Operations. The performance parameters MAE, R-square are given as:

$$MAE = \sum_{i=1}^{n}(target_i - predicted_i)\,/n$$

$$R - SQUARE = 1 - \left(\frac{\sum_{i=1}^{n} target\ _i - predicted\ _i)^2}{\sum_{i=1}^{n} target\ _i - predicted\ _i}\right)$$

Prediction result: The performance of BPNN, ELNN and JNN in the process of predicting the target is shown in the Fig. 3 to 5, respectively.

Comparative study of NN performance: This study aims at comparing efficiency and accuracy of BPNN, ELNN and JNN. The MAE, R-square and CPU usage time for the above three NN are given in Table 2. It shows that JNN achieves better accuracy and efficiency.

Table 1: Data set format for NN training

Input feature 1....	Input feature m	Output data
X (1)…	X (n)	X (n+1)
X (2)…	X (n+1)	X (n+2)
…	….	…..
X (t-2)…	X (t-m-1)	X (t-1)
X (t-1)…	X (t-m)	X (t)

Table 2: Performance table of NN by back propagation algorithm

Parameter	BPNN	Elman	Jordon
MAE	0.0376	0.0235	0.0027
R-square	0.9496	0.9711	0.9956
CPU time	8 sec	6 sec	5 sec

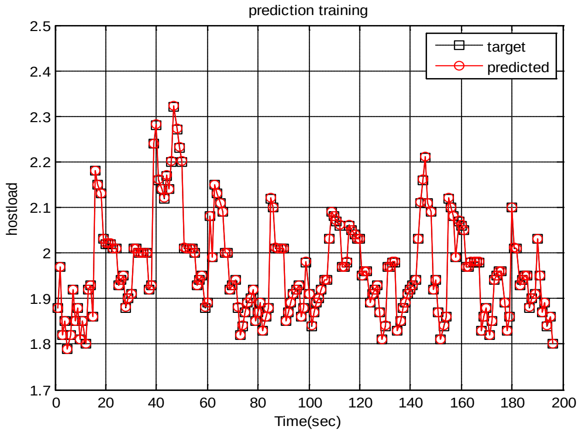

Fig. 3: BPNN prediction

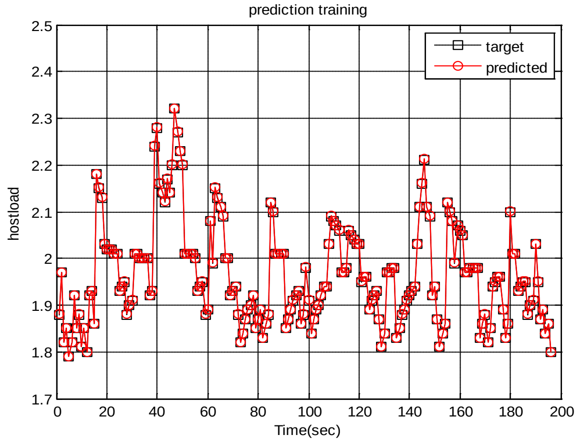

Fig. 4: ELNN prediction

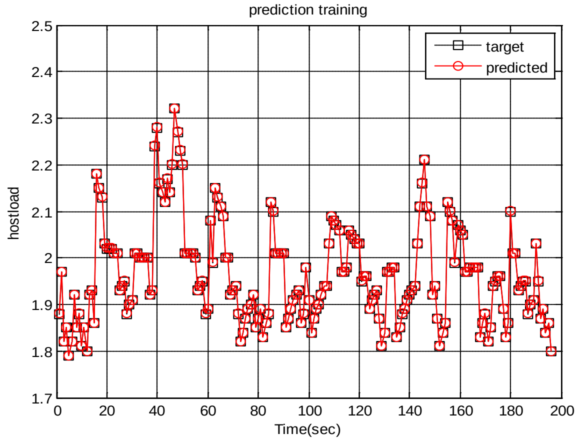

Fig. 5: JNN prediction

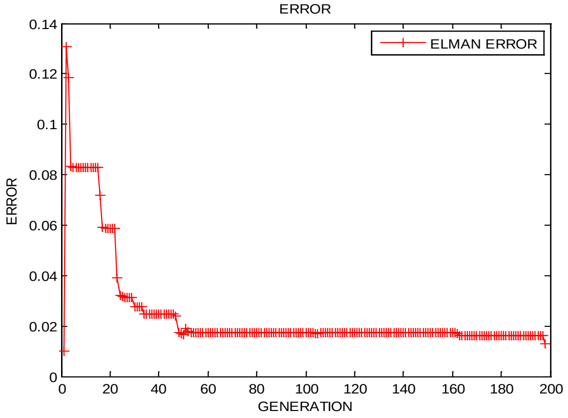

Fig. 6: GA-BPNN error

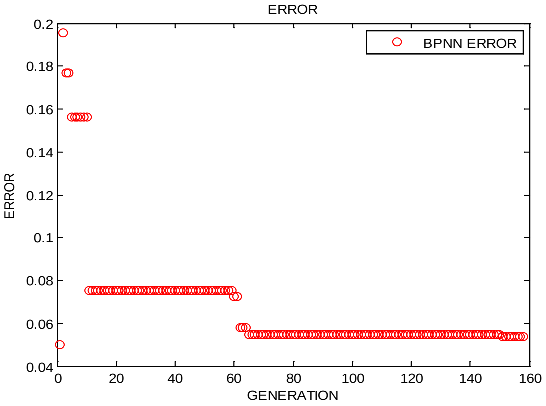

Fig. 7: GA-ELNN error

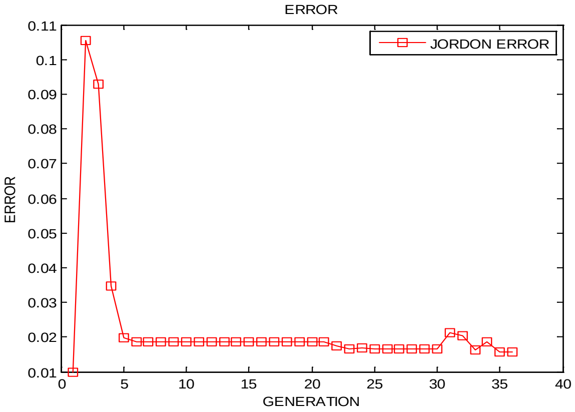

Fig. 8: GA-JNN error

Table 3: Performance table for NN by weight optimized genetic algorithm

Parameter	BPNN	Elman	Jordon
MAE	0.0105	0.0062	0.0005
R-square	0.9623	0.9843	0.9962
CPU time	6 sec	3 sec	2 sec

The Error performance of genetically weight optimized BPNN, ELNN and JNN in prediction process is shown in Fig. 6 to 8, respectively.

Comparative study of GA-ANN performance: The performance Comparison of genitally weight optimized BPNN, ElNN and JNN is tabulated in Table 3. This shows that R-square value of JNN and ELNN are near, but the consumption of CPU time is concerned JNN has taken less comparatively.

CONCLUSION

This study proposes architecture for IaaS cloud computing platform in which VM allocation is done through genetically weight optimized neural network. We use host load of each system as resource prediction parameter for identify a system to launch VM. We analyses the prediction performance of BPNN, ELNN and JNN. It is found that recurrent network JNN gives better result than ELNN and BPNN. The number of generations taken by the weight optimized GA-JNN and number of epochs taken by the Back Propagation trained-JNN is less than other neural networks. The GA-JNN is best when accuracy is concerned, both the weight optimized ELNN and JNN neural networks are best fit for the prediction of resources which is proved by R-square value, when Efficiency is concerned JNN trained by GA gives better Efficiency. This prediction certainly helps VM allocator to select a right system for launching VM.

RECOMMENDATIONS

We have been developing strategic decision making algorithm, which would use host load prediction as one of the prime parameter along with other dynamic parameters of cloud hub to create and allocate VM for IaaS request. We prepare to evaluate the performance of proposed architecture in Microsoft Cloud tool.

REFERENCES

Alaknantha, E., 2005. A neural network based predictive mechanism for available bandwidth. Proceeding of the 19th IEEE International Parallel and Distributed Processing Symposium, pp: 33a.

Bohlouli, M., 2008. Grid-HPA: Predicting resource requirements of a job in the grid computing environment. J. World Acad. Sci. Eng. Technol., 42: 2008.

Che, X.L., 2010. A nu-support vector regression based system for grid resource monitoring and prediction. Acta Automat. Sinica, 36(1).

Dinda, P.A., 2001. Host load prediction using linear models. J. Clust. Comput., 3(4): 265-280.

Doulamis, N.D., A.D. Doulamis, A. Panagakis, K. Dolkas, T.A. Varvarigou and E.A. Varvarigos, 2004. A combined Fuzzy Neural model for nonlinear prediction of 3-D rendering workload in grid computing. IEEE T. Syst. Man Cy. B, 34(2): 1235-1247.

Duy, T.V.T., Y. Sato and Y. Inoguchi, 2009. Improving accuracy of host load predictions on computing grids by artificial neural network. Proceeding of IEEE International Symposium on Parallel and Distributed Processing, pp: 23-29.

Ioan, I., 2004. The optimization of feed forward neural structure using genetic algorithms. Proceeding of the International Conference on Theory and Applications of Mathematics and Informatics, pp: 223-234.

Jeffrey, E., 1990. Finding structure in time. Cognitive Sci., 14: 179-211.

Jin, H., 2007. Using NARX neural network based load prediction to improve scheduling decision in grid environments. Proceeding of the IEEE International Conference Natural Computation (NC, 2007).

Liang, H., C. Xi-Long and Z. Si-Qing, 2012. Online system for grid resource monitoring and machine learning-based prediction. IEEE T. Parall. Distr., 23(1).

Ming, W., 2006. Grid harvest service: A performance system of grid computing. J. Parallel Distr. Com., 66(10).

Neto, L.B., P.H.G. Coelho, J.C.C.B. Soares de Mello, L.A. Meza and M.L. Fernandes Velloso, 2004. Flow estimation using an Elman networks. Proceeding of the IEEE International Joint Conference on Neural Networks.

Zhen-Guo, C., 2011. Feed-forward neural networks back-propagation learning algorithm. Int. J. Innov. Comput. I., 7(10): 5839-5850.