Proficient Decision Making on Virtual Machine Creation in IaaS Cloud Environment

Radhakrishnan Ayyapazham and Kavitha Velautham
Department of Computer Science, Anna University, India

Abstract: *Cloud computing is a most fascinated technology that is being utilized by IT companies to reduce their infrastructure setup cost by outsourcing data and computation on demand. Cloud computing offer services in three basic models such as SaaS, PaaS and Infrastructure as a Service (IaaS). Where IaaS is one of the fundamental cloud service model in which cloud provider offers Virtual Machines (VMs) as resources to cloud customers through virtualization. The VMs act as dedicated computer system to consumers which are created on physical hosts of cloud provider. Making decision of physical host selection for VMs creation is a challenging task for cloud provider. Any deficiency of this selection causes VMs migration in middle of computation or restart computation from the scratch; these would sternly affect profit and trust of cloud provider. In this paper, we proposed a novel methodology to handle VMs creation and allocation for IaaS service. The proposed methodology employs a genetically weight optimized neural network component in each host to predict their near future availability during VMs creation. We analyses the host load prediction performance of various neural networks through real time host load values. Also we proposed a proficient decision making algorithm named Future Load Based Virtual machine Creation (FLBVC) to choose appropriate launching hosts for VMs. The performance of our methodology is validated using CloudAnalyst tool. The results demonstrated that our proposed approach reduces response time of cloud customers and rental cost of VMs.*

Keywords: *IaaS, VMs, jordon neural network, genetic algorithm, service level agreements, FLBVC.*

Recieved October 20, 2013; accepted July 2, 2014

1. Introduction

Cloud computing is a service oriented architecture, where computing resources are provided to cloud customer via internet from cloud provider. The unique features of cloud computing are dynamic scaling based on application requirements, usage based pricing, fast service provisioning, multiple tenants coexisting on the same infrastructure. These features attracted the attention of IT companies, they reduces their expenses by outsourcing their data and computation on demand [6]. Cloud computing provides three major services such as SaaS, PaaS and Infrastructure as a Service (IaaS). SaaS provides well defined applications, where users can use the applications through internet without installing and maintaining. PaaS is a platform service that provides facility to customers to build and deploy their applications. IaaS service provides computing infrastructure, where customer can rent machine instances according to their requirement specification. These instances behave like dedicated system that is controlled by the customer, who therefore fully responsible for their operation. These services operate on pay-per-usage model ensuring that the consumers pay only what they are using. The cloud can be deployed at any one of three fundamental models namely private, public and hybrid cloud. Despite of these potential benefits and features IT companies are reluctant to do this business due to outstanding issues [18, 20].

The main focus of our research work is on IaaS service model. As all cloud services are running on top of IaaS, it acts as a foundation for cloud computing. Any enrichment in IaaS would automatically reflect other service models too. One of the major issues in IaaS service is Virtual Machines (VMs) creation and allocation. In IaaS service VMs are created on physical machines as machine instances, they are provided as resources to IaaS customers on demand. At this juncture the objective of the provider is profit oriented handling of requests and maintains their credibility. The expectation of customer is to get resources with low cost, complete computation on time without any hitch. The major shortfalls of these expectations are VMs migration on middle of the computation or VM failure due to heavy load of hosts where VMs are running. These letdown increase completion time or may start the customer application from the scratch. These factors are severely affecting the profit of both parties as well as diminish trust on IaaS provider [13].

In this paper, a novel methodology for IaaS service is presented that takes into account the resources near future availability based VMs creation and strategic allocation. The prediction of future availability of resources can be extremely useful for many purposes. First, resources 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. It waste the valuable CPU cycles and increases application span. The prediction can allow the resource selector to choose resources that are least likely to become unavailable and avoid application restart. This can increase the reliability of the system. This research work contains the utilization of neural network and Genetic Algorithm (GA) in cloud resource near future load prediction. The resource availability predictor is framed by Artificial Neural Network (ANN) with GA, which forecasts the near future availability of resources while VMs creation process is take place in IaaS service. It can help resource selector to make better decisions about resource selection. Resource prediction is based on the resource monitoring that provides the historical data which describes the past experience. Each computing resource consists of a queue data structure to keep track of its past experience. This is the key parameter for future availability prediction.

We analyses the prediction performance various neural networks by optimizing their weights using GA. Based on future load prediction of resources we proposed a proficient decision making algorithm Future Load Based Virtual machine Creation (FLBVC) for VMs creation. The FLVBC is affixed with each cloud computing resources that would be called during VMs creation. The performance of our proposed methodology is validated in CloudAnalyst tool. The results exhibit that our novel methodology significantly reduces VMs rental cost as well as task completion time of customers by reducing data centers processing time.

2. Related Works

Liang *et al.* [16] proposed the design and implementation of grid resource monitoring and its load prediction, whereas radial basis function and back propagation neural networks are projected for the prediction of grid resources. Duy *et al.* [11] employed to improve the accuracy of host load prediction. They used back propagation neural networks to train the collected load traces that give consistence performance with low overhead in load balancing. Doulamis *et al.* [10] developed neural network with constructive algorithm for the prediction of workload. The workload predictor is based on combined fuzzy classification and neural network model. The fuzzy classification used to increase the prediction accuracy and constructive algorithm is employed to train the neural network in order to estimate network weights and size simultaneously. Dinda and Hallaron [9] studied different linear series models including various autoregressive methodologies for predicting future loads from 1 to 30 seconds. An idea of active database and centralized history maintenance is proposed by Bohlouli and Analoui [4]. The centralized history maintenance is used to maintain the resource details used by the jobs and active database stores the attributes of resources as well executed job. This information is used to predict the resource requirement of upcoming jobs. The prediction process is done by recurrent neural networks. Alaknantha *et al.* [1] presented a model for the prediction of network traffic using bandwidth data. In this approach multi layer perceptron neural network is used with the ability to extract patterns and detect available bandwidth. Wu and Sun [23] described the mean based method for the prediction of resource availability. The parameters used for prediction includes arrival rate of jobs, utilization of machine, standard deviation of service time and computing capacity of machine. Das and Choudhury [8] developed prediction model through feed forward and recurrent Elman network. They used weekly sales data of footwear shop and the information about the seasonality of sales process for the prediction of forthcoming sales. Baptista and Dias [2] described detailed survey on artificial neural network training tools, whereas they listing and explaining various neural network tools and its characteristics and terms of use. Che *et al.* [7] analyses the advantages and characteristics of genetic algorithm and backtracking neural networks to cope with weight adjustment. Mahalee *et al.* [17] describes the performance of existing load balancing algorithm in cloud environment, they concluded throttled load balancing algorithm works more efficiently in terms of cost and request processing of data centers. James and Verma [14] proposed weighted active load balancing algorithm for cloud computing environment, whereas the VMs are assigned a varying amount of the available processing power to the individual application services. Kun *et al.* [15] proposed a framework for predicting task execution time that is used for task scheduling and resource allocation in cloud computing environment.

Habib *et al.* [13] analysed the factors for maintain trust in cloud computing. The reliability of offering resource to cloud customer is a major concern to uphold the trust. A survey reveals that the provider promise of high availability of resources is futile. TaheriMonfared and Jaatun [19] developed a model for identifying and blocking infected hosts from IaaS service. The prediction is done by study the profile of infected host and compares it to other working host profile. It helps to provide reliable IaaS service to customers. Belognazov *et al.* [3] developed an architectural framework and principles for energy-efficient cloud computing that reduces the operational cost of data centers. Buyya *et al.* [5] proposed market oriented cloud computing architecture that maps customer request in to appropriate cloud provider ensuring quality of service and minimum cost. Ferrer *et al.* [12] developed multi cloud architecture that consists of resource broker decision making process. The trust, cost and risk in provider hub are the key parameter for resource broker to select a provider. Weickremasinghe *et al.* [21] construct a graphical simulation tool CloudAnalys that is built on top of the CloudSim toolkit, developed by the Cloud computing and distributed system laboratory at Melbourne University. It enables to make simulation environment for cloud computing.

3. Materials and Methods

3.1. IaaS Cloud Architecture

The proposed novel architecture of IaaS service is depicted in Figure 1, which consists of many elegant components with defined functionalities. The behaviour of each component is given below.

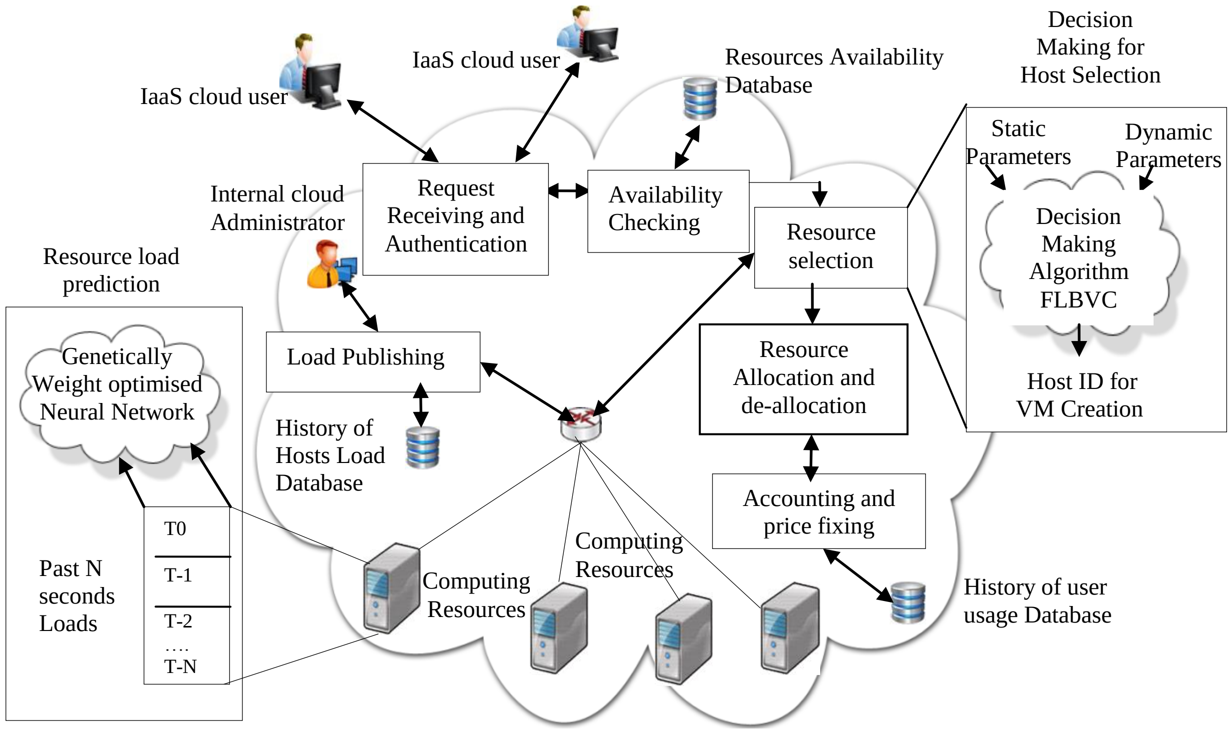

Figure 1. IaaS service architecture with FLBVC.

- *Request Receiving and Authentication:* It receives request from users and verify the authenticity of users for providing service.
- *Resource Availability Checking:* It verifies availability of requested resource from cloud hub. If requested reosurce is not available then request is redirected to another cloud hub as per SLA.
- *Resource Selection:* It intiates neural network load prediction process of each host, and finally chooses a suitable host for VMs creation.
- *Resource Load Prediction Process:* This process is embedded in all hosts of cloud hub, the main role of this process is to predict the near future load of each host based on its past load history. It is made by genitically weight optimized neural nerwork.
- *Load Publishing:* The predicted load of each host is published to internal cloud administrator. This notification would help the administrator to know the status of each resourses.
- *Decision Making:* This activity is part of resource selection unit to choose reliable hosts to lauch VMs. The proposed FLBVC algorithm is instrumental for proficient decision making.
- *Resource Allocation and Deallocation:* This unit deals with launching of VMs for requests after ensuring compliance of SLA. The VMs would be revoked once usage is over.
- *Accounting and Price Fixing:* This unit keeps usage information about allocated VMs as well as to calculate the usage cost as per the cloud policy.

3.2. System Work Flow

The work flow of the novel architecture is presented as sequence diagram in Figure 2. The detailed descriptions of major activities are as follows:

1. The IaaS requester log on by valid authentication and submit their request details.
2. The requested resources availability is checked in cloud hub database.
3. If availability exits then the neural network based resources load prediction process is initiated.

4. The predicted load of each host is brought to the attention of hub administrator as well as resource selection unit for VMs creation.
5. Resource selection unit call decision making algorithm for identifying suitable host by providing dynamic and static parameters such as resources predicted load, requested type of virtual machine and user category details.
6. Decision making process is done by FLBVC that identifies targeted physical host for virtual machine creation and notifies the same to resource selection unit.
7. Virtual machine created on targeted host and allocate to customer, the update is made on resource availability database and accounting unit.
8. Customer received requested resource as virtual machine instance, after usage relinquish request send to cloud hub.
9. Revoke allocated resource from customer and update to accounting unit and resource availability units.
10. Cost of resource usage is intimated to customer for payment.

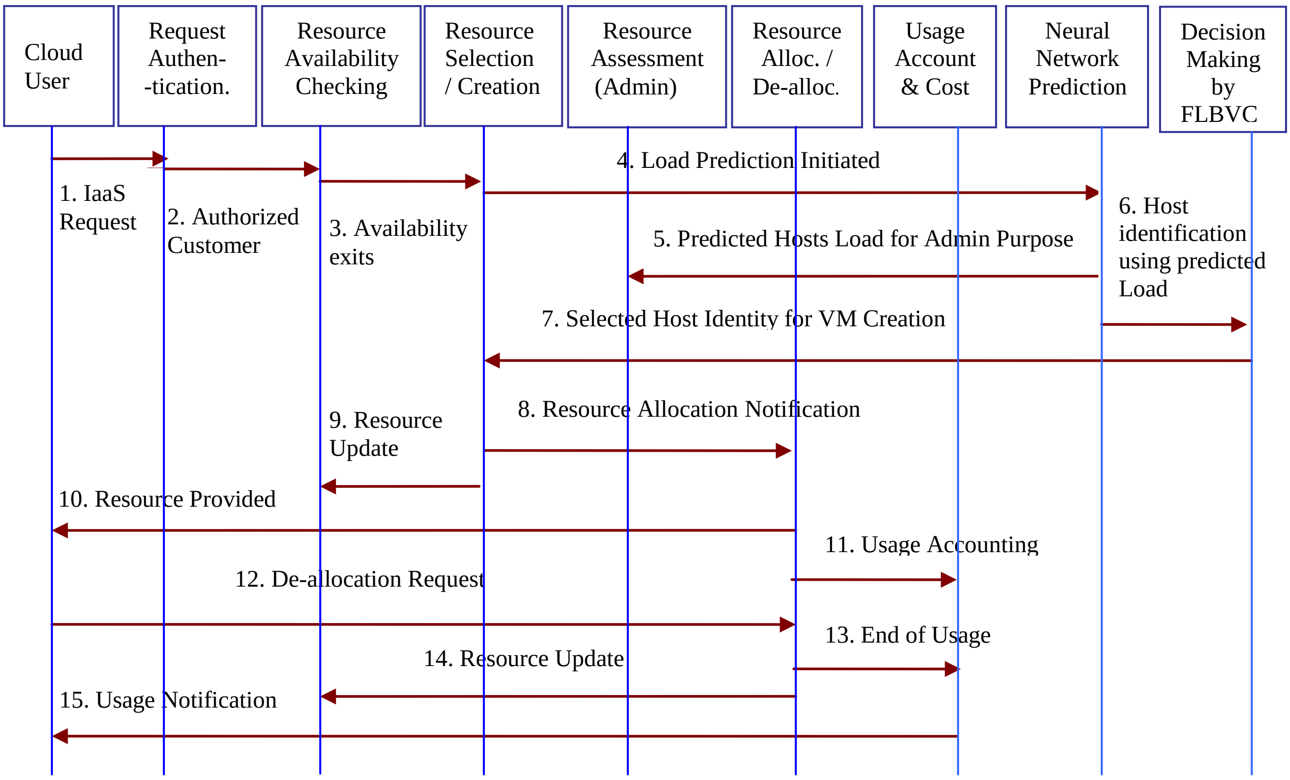

Figure 2. Sequence diagram of IaaS Architecture activities.

3.3. Weight Optimization of Neural Network Using Genetic Algorithm

In neural network weights are playing vital role to produce good results. We optimize the weights by GA to minimize the learning error. The first step, weights are encoded into chromosome format and the second step is to define a fitness function based on the error in neural network training, that calculates the fitness of each chromosomes. The selected pair of fitted chromosomes is send to combination process. The combination process performs crossover and mutation to produce the new chromosomes. A crossover takes two parent chromosomes and creates a single child with genetic material from both parents. A mutation operator selects a gene in a chromosome and adds a small random value. The third step is to replace the initial chromosomes with newly generated chromosomes. The optimized weight obtained from GA process is given as new weight for neural network training. These processes are repeated until minimum value of error is occurred in neural network training.

3.4. Virtual Machine Allocation Model

It assumes that a customer request for VMs from provider is by agreeing the SLA executed between them. An IaaS provider can posses multiple cloud hub in geographically different location and can forward the request from one hub to another if necessary. In the proposed IaaS architecture VMs can be provided in three different types. The VMs are classified according to computation power sharing from total computation power of its physical host. The VMs classification and their cost are specified in Table 1.

Table 1. Virtual machine classification.

SL.NO	VM TYPE	Price per Hour
1	Small	0.08$
2	Standard	0.3$
3	High	0.6$

The customer is also classified according to their usage. The category of customers is obtained from cloud database, which keeps the usage information as well as contract details of customers. When request arrives for VMs, the respective grade of requested customer is obtained from cloud database. The customer classification is denoted in Table 2.

Table 2. Taxonomy of cloud customers.

SL.NO	Type of Customer	Grade
1	Contracted User	A
2	Above 50 Hrs Usage	B
3	Below 50 Hrs Usage	C

The customer grade and requested VMs types are vital parameters for decision making process to choose reliable launching hosts of VMs.

3.5. Algorithm

The main objective of our proposed algorithm is to identify suitable physical host for VMs creation and strategic allocation to customers, which minimizes VMs failure and migration due to hefty work load of VMs running host. The breakdowns of VMs during service severely affect the trust as well as profit of IaaS provider.

3.5.1. Proposed Algorithm FLBVC

We propose the algorithm FLBVC for efficiently handling IaaS request in cloud hub. This algorithm is designed to choose appropriate physical host to create VMs based on its near future load. The predicted load of a host is represented between the values 0 and 1. If the predicted load of a host is 0 that seems the host is idle. The load value 1 indicates the host is running with its full capacity, in between values points out the load level of host. This algorithm classifies the hosts of cloud hub into three categories, heavy, medium and low according to the predicted load level. The load prediction process is done through genetically weight optimized neural network.

The algorithm FLBVC consists of two sub functions, NN-PRED (N) and Host-Class (Load). The sub function NN-PRED (N) is used to predict the near future load of each host. The parameter 'N' is set with predefined value that specifies how long past history is taken for prediction. The function Host-Class (Load) performs the categorization of hosts according to parameter 'Load'. The physical hosts in the cloud hub are having its identity that is represented by variable Host-ID. The pseudo code of the algorithm is given below.

Algorithm 1: Future Load Based Virtual Machine Creation (FLBVC).

```
Input : VM-Type, Cus-ID, N
Output: Host-ID
FLBVC (VM-Type, Cus-ID, N)
 {
/* The array size 'n' depends on number of physical host in
cloud hub */
Array: Heavy [n][2], Medium [n][2], Low [n][2]
/*Columns contains Host-ID and Load, all field are set to null
initially*/
Integer: Load; Character: Cus-Grade
1 If ( Cus-ID is in blocked list of service) {
2        service request is dropped, break; }
3        Else {
 /*The customer gets any one of the grade among A,
 B and C. */
4 Cus-Grade = Obtain customer grade from cloud
 database through Cus-ID;
5        loop {
6        For each Physical host in cloud hub {
7 Load = Call NN-PRED (N)
8 Call Host-Class (Load) } } End Loop
9        SORT Arrays Low, Medium and Heavy
 according to load value;
 /* checking whether cloud hosts are fully loaded */
10 If (Low and Medium Array are null &&
 VM- Type == 'Heavy' or 'Standard')
11       { Forward request to another cloud hub;
12 break ; }
13 Else {
14       Switch (VM-Type)
15       Case 'High':
16 IF (Low Array != null) {
17       Return Host-ID of first host in Low Array;
18       break } Else
19       Return Host-ID of first host in Medium
 Array;
20              break;
21       Case 'Standard':
22       If ( Cus-Grade== 'A' or 'B') {
23       If ( Low Array != null) {
24 Return Host-ID of first host in Low Array;
25       break; } Else
26       Return Host-ID of less load host in Medium
 Array; }
27       break;
28       Case 'Small':
29       If (Low and Medium Array are null) {
30 Return Host-ID of lowest load host in Heavy
 Array;
31       break; }
32       If ( Cus-Grade == 'A') {
33 Return Host-ID of first host in Low Array;
34       break; }   Else
35       Return Host-ID of lowest load host in
 Medium Array;
36       break;
37 } } }
```

- SubFunction NN-PRED(N)

Algorithm 2: Finding near future load of physical hosts.

```
NN- PRED (N) {
```

/ Queue is attached with each physical host to monitor its past loads */*
1 Fetch past N seconds load of host from queue data structure.
2 The obtained loads are trained by genetically weight optimized neural networks to predict its near future load;
3 Return the predicted load to main function;
4 }

- *SubFunction Host-Class(Load)*

Algorithm 3: Classification of hosts according to predicted future load.

Host-Class (Load) {
/ categorized hosts according to their load value */*
1 If (Load value is less than 40 %) {
2 The Host-ID and Load are move to array Low;
3 break; }
4 Else {
5 If (Load value is between 40 % and 70 %) {
6 The Host-ID and Load are move to array Medium;
7 break; }
8 Else
9 The Host-ID and Load are allot to array Heavy;
10 Return }}

The pseudo codes of the above algorithms are converted into JAVA code during implementation in CloudAnalyst tool to evaluate its performance.

3.6. Profit Calculation Model for Provider

The profit calculation model helps the IaaS provider to calculate the cost benefit of rendering infrastructure service. The major shortfall of profit is paying penalty due to violation of SLA [22]. The reasons for SLA violation in IaaS service are VMs failure during middle of execution as well as VMs migration. The root cause of these problems is wrongly chosen host for launching VMs. Our proposed methodology addresses this issue firmly. The profit calculation is as follows.

Let K denotes the number of customer request and k specify customer request id. Let t indicates type of the VM and i denotes its id. The VMs cost is depends up on its type t. The specific VM_i with its type VM_{it} has priced $CostVM_{it}$. Let $VMusage_{it}^{k}$ be the usage hours of allotted VM_{it} to K customers and $Infra.Cost_{it}^{k}$ be the cost of expenditure by serving the customer K with VM_{it}^{k}. The total revenue $\sum_{k=1}^{K} Revenue_{it}^{k}$ earned by the provider for serving K number of customer request is specified in the following Equation 1.

$$\sum_{k=1}^{K} Revenue_{it}^{k} = \sum_{k=1}^{K} CostVM_{it}^{k} \times \sum_{k=1}^{K} VMusage_{it}^{k} - \sum_{k=1}^{K} Infra.Cost_{it}^{k} \quad (1)$$

A customer request k, the service cost $CostVM_{it}$ depends up on the virtual machine type as well as customer account type. The both contract and long time relation customers are having offer in their usage cost. The $Infra.Cost_{it}^{k}$ indicate expenditure for serving K requests that depends on the $PenaltyCost_{it}^{k}$ as well as $InternalCost_{it}^{k}$ as specified in Equation 2.

$$Infra.Cost_{it}^{k} = PenaltyCost_{it}^{k} + InternalCost_{it}^{k} \quad (2)$$

The SLA violation leads to penalty. The profit of the provider is significantly reduced by paying penalty for their deficient service. The penalty cost is defines in Equation 3.

$$PenaltyCost_{it}^{k} = \alpha + \beta(RequestedVMtype) \times delayTime_{it}^{k}(RequestedVMtype) \quad (3)$$

The penalty cost depends upon the delay in providing requested virtual machine. The delay time is calculated by the time exceed against the agreed response time mentioned in SLA. The penalty rate specified as β and constant value α [24]. The delay time and penalty cost are varied upon requested virtual machine type. The internal cost $InternalCost_{it}^{k}$ denotes electricity cost and cloud hub operating expenses including maintenance. This model is support to the IaaS provider to visualize the profit after serving K service requests.

3.7. Experimental Setup

The performance results of our proposed methodology in IaaS service is obtained from two set of experiments. The first experiment is done for evaluate the prediction performance of genetically weight optimized neural networks Back Propagation Neural Network (BPNN), Elam Neural Network (ELNN) and Jordan Neural Network (JNN). The second experiment is carried out in IaaS cloud environment to validate the performance of our novel algorithm FLBVC. We compared the performance of our algorithm with the existing algorithm in IaaS cloud environment. In subsequent section, we explain our experimental methodology in load prediction followed by cloud architecture setup in CloudAnalyst tool.

3.7.1. Neural Network Prediction Setup

The load of a grid computing node in internet is the most appropriate resource information. This information is highly suitable to built prediction model. For host load dataset, we have taken load dataset *mystere10000.dat* from *http://people.cs.uchicago.edu*, a load trace of grid workstation node. The 200 samples of load were sequentially taken to form experimental dataset. The transformation of selected dataset for neural network training is specified in Table 3. In neural network training five input nodes, five hidden nodes and one output node are formed. The performance of prediction process is calculated by training CPU time that measures efficiency, and Mean Absolute Error (MAE) to measure accuracy.

Table 3. Neural network training dataset model.

Input Data 1 Input Data M	Predicted 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)

To assess the correctness of fitting between target and prediction, the R-square statistic measurement is used. In weight optimization by GA, two site crossover is used and the chromosomes with least error value are taken for the genetic operations.

3.7.2. Cloud Architecture Setup in CloudAnalyst Tool

Cloud architecture setup is made in CloudAnalyst toolkit that provides simulation environment for IaaS cloud computing service. It also facilitates to customize the functionality of core IaaS system components such as datacenters, VMs creation, resource allocation policies and request making pattern. Our proposed methodology is stuffed to these components to redefine its functionality. In our experimental, we configured two heterogeneous datacenters and ten user hubs. Each datacenter is having unique identifier and is located in different geographical location. CloudAnalyst supports VMs provisioning at two levels, one is at host level, and another one is VMs level. At the host level VMs provisions are made to specify how much of the overall processing power of host will be assigned to each virtual machine. At the VMs level, the VMs assign a fixed amount of the available processing power to the individual application services. The proposed architecture chosen the first level of VMs provisioning where VMs share the overall processing power of host based on their type. The setup for creating IaaS cloud environment in CloudAnalyst tool is specified in Table 4.

Table 4. Parameters setup in CloudAnalyst tool.

Simulation Setup in CloudAnalyst Tool	
Number of User Base	10
Number of Data centers	2
Data centers OS	Linux
Data centers host architecture	X86
Physical Hardware Unit	2
Number of Processors	4
RAM (GB)	8
Storage capacity of hosts (TB)	1
Service Broker Policy	Optimal Response Time
VM Load Balancer	Throttled
Request per user / hour	60
Data size per request (Bytes)	100
Internet Characteristics	Default Setup

The simulation is first run with setting as specified in Table 4, next the proposed methodology is adapted and run the simulation. The results of the simulation are analyzed in next section.

4. Results and Discussions

The graphs in the following Figures 3 and 4 depict the error value in each generation and number of generation taken to optimize the weights using GA in BPNN and ELNN. The BPNN took 160 generation and ELNN takes around 200 generation.

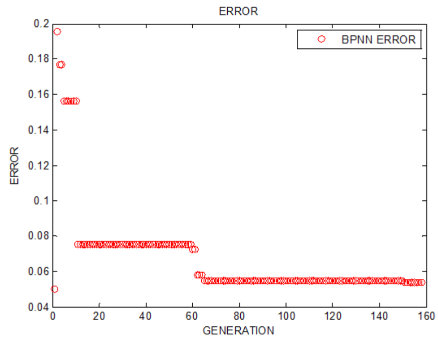

Figure 3. GA-BPNN error.

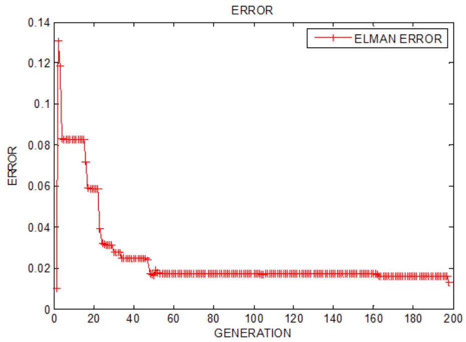

Figure 4. GA-ELNN error.

The graph in the following Figure 5 represents error versus number of generation taken by JNN. The JNN optimizes the weights in less than 40 generation. It is comparatively lower than other neural networks.

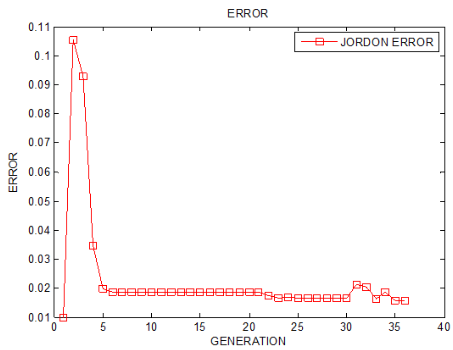

Figure 5. GA-JNN error.

The performance of neural network prediction is measured by MAE, R-Square and CPU time. The comparisons of genetically versus default weight optimized neural network prediction of BPNN, ELNN and JNN are shown in Table 5.

Table 5. Comparisons of default versus genetically weight optimized neural networks prediction results.

NN Model	PARAMETERS					
	Default weight			Genetically Optimized weight		
	MAE	R-Square	CPU Time	MAE	R-Square	CPU Time
BPNN	0.037	0.949	8 sec.	0.010	0.962	6 sec.
ELNN	0.023	0.971	6 sec.	0.006	0.984	3 sec.
JNN	0.002	0.995	5 sec.	0.0005	0.996	2 sec.

The experimental results in Table 5 shows that the performance of genetically weight optimized neural networks are well compare to default weight assigned neural networks in prediction. The genetically weight optimized ELNN and JNN are very close in accuracy of prediction but considering CPU time taken, JNN is less compare to ELNN. It took only two seconds to predict 200 samples of load. Based on the experimental result, we have chosen genetically weight optimized JNN for cloud host load prediction process.

The simulation result screen of CloudAnalyst tool contains map of the world which is divided into six regions. The simulation setup encloses ten user hubs in different region and two data centers to cater to the needs of user hubs. The user hubs, which are distanced from data centers get high network latency that reflects in response time. The user hubs response time comparison of existing Throttled algorithm based VMs allocation as well as proposed FLBVC is depicted in Figure 6, whereas the user hubs 4, 5 and 6 are having high network lately compare with others.

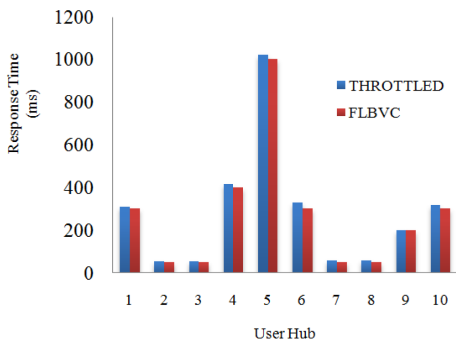

Figure 6. Response time comparison of throttled and FLBVC.

The Throttled algorithm performance has proved as best among all existing load balancing algorithm in cloud environment [17]. The result shows that the FLBVC immensely reduces the response time of user hubs compared to Throttled algorithm. The Figure 6 demonstrates the response time comparison of existing and proposed methodology.

In data centres perspective, the simulation result shows that the overall processing time of data centers are significantly reduced by FLBVC compared with Throttled algorithm. It reduced processing time of data centers that ultimately minimizes the rental cost of VMs. The comparison is presented in Table 6.

Table 6. Performance comparison of throttled and FLBVC.

PARAMETERS	THROTTLED	FLBVC
Average datacenters processing time	2.01(ms)	1.31(ms)
Max. Processing time of request	2.87(ms)	2.33(ms)
VM Cost	5.98 $	5.1.$

5. Conclusions and Future Works

Cloud computing is a great boon to IT industries to solve their problems in less cost and low time. It is scalable as per the demand of customers that can be accessed via internet. One of the fundamental service models of cloud is IaaS, which offers remote delivery of entire computer infrastructure as VMs. The major obstacle in this service model is to create reliable VMs for customer requests. Any deficiency in VMs creation cause major shortfall in profit for both provider and customer. We proposed a novel methodology to address this issue strongly. The methodology is embedded in algorithm FLBVC, whereas VMs creations on physical hosts are based on its near future load. It helps the provider to render reliable VMs and escape from SLA violation. The future load prediction process is fabricated by genetically weight optimised JNN. Our novel algorithm FLBVC is tested in CloudAnalyst toolkit to evaluate its performance, whereas FLBVC performance is compared with existing Throttle algorithm. The performances of these algorithms are evaluated by parameters such as response time of customer application and data centers processing time. The simulation result demonstrates that FLBVC provides less response time and reduce data centers processing time compare with Throttled. It increases the profit of provider and reduces the rental cost of customer.

In continuation of the research, we will concentrate on SLA slip against our methodology. We also explore the ways to utilize our methodology in other IaaS activities such as VMs migration and service broker policy refinement.

Acknowledgement

This research work is supported by DST-SERB-(INDIA) Fast Track Scheme for Young Scientist /F/0590/2012-13 dated 09.05.2012. The authors would like to thank the editor and anonymous reviewers for giving valuable comments and fruitful suggestions.

References

[1] Eswaradass A., Sun X., and Wu M., "A Neural Network Based Predictive Mechanism for Available Bandwidth," *in Proceeding of 19th International Parallel and Distributed Symposium*, Colorado, 2005.

[2] Baptista D. and Dias F., "A Survey of Artificial Neural Networks Training Tools," *Neural*

Computing and Applications, vol. 23, no. 3, pp. 609-615, 2013.

[3] Beloglazov A., Abawajy J., and Buyya R., "Energy Aware Resource Allocation Heuristics for Efficient Management of Datacentres for Cloud Computing," *Journal of Future Generation Computer System*, vol. 28, no. 5, pp. 755-768, 2012.

[4] Bohlouli M. and Analoui M., "Grid-HPA: Predicting Resource Requirements of a Job in the Grid Computing Environment," *International Journal of Electrical and Computer Engineering*, vol. 3, no. 3, pp. 137-141, 2008.

[5] Buyya R., Pandey S., and Vecchiola C., "Market Oriented Cloud Computing and the Cloudbus toolkit," *in Proceeding of the 1st International Conference on Cloud* Computing, Beijing, pp. 24-44, 2009.

[6] Buyya R., Yeo C., Venugopal S., Broberg J., and Brandic I., "Cloud Computing and Emerging IT Platforms; Vision, Hype and Reality for Delivering Computing as 5th Utility," *Journal of Future Generation Computer System*, vol. 25, no. 6, pp. 599-616, 2009.

[7] Che Z., Chiang T., and Che Z., "Feed-Forward Neural Network Training: A Comparison Between Genetic Algorithm and Back-Propagation Learning Algorithm," *International Journal of Innovative Computing, Information and Control*, vol. 7, no. 10, pp. 5839-5858, 2011.

[8] Das P. and Choudhury S., "Prediction of Retail Sales of Footwear using Feed Forward and Recurrent Neural Networks," *Journal of Neural Computing and Applications*, vol. 16, no. 4, pp. 491-502, 2007.

[9] Dinda P. and O'Hallaron D., "Host Load Prediction Using Linear Models," *Journal of Cluster Computing*, vol. 3, no. 4, pp. 265-280, 2000.

[10] Doulamis N., Doulamis A., Panagakis A., Dolkas K., Varvarigou T., and Varvarigou E., "A Combined Fuzzy-Neural Network Model for Nonlinear Prediction of 3-D Rendering in Grid Computing," *IEEE Transaction on Systems Man and Cybernetics*, vol. 34, no. 2, pp. 1235-1247, 2004.

[11] Duy T., Sato Y., and Inoguchi Y., "Improving Accuracy of Host load Predictions on Computing Grids by Artificial Neural Network," *International Journal of Parallel, Emergent, and Distributed System*, vol. 26, no. 4, pp. 275-290, 2011.

[12] Ferrer A., Hernndez F., Tordsson J., Elmroth E., Ali-Eldin A., Zsigri C., Sirvent R., Guitart J., Badia R., Djemame K., Ziegler W., Dimitrakos T., Nair S., Kousiouris G., Konstanteli K.,Varvarigou T., Hudzia B., Kipp A.,Wesner S., Corrales M., Forg N., Sharif T., and Sheridan C., "OPTIMIS: A Holistic approach to Cloud Service Provisioning," *Journal of Future Generation Computer System*, vol. 28, no. 1, pp. 66-77, 2012.

[13] Habib S., Hauke S., Ries S., and Muhlhauser M., "Trust as a Facilitator: A Survey," *Journal of Cloud Computing: Advances, Systems and Applications*, vol. 1, no. 19, pp. 1-18, 2012.

[14] James J. and Verma B., "Efficient VM load Balancing algorithm for a Cloud Computing Environment," *International Journal of Computer Science and Engineering*, vol. 4, no. 9, pp. 1658-1663, 2012.

[15] Gao K., Wang Q., and Xi L., "Reduct Algoritham Based Execution Times Prediction in Knowledge Discovery Cloud Computing Environment," *International Arab Journal of Information Technology*, vol. 11, no. 3, pp. 268-275, 2014.

[16] Hu L., Che X., and Zheng S., "Online System for Grid Resource Monitoring and Machine Learning-Based Prediction," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 1, pp. 134-145, 2012.

[17] Mahalee H., Kaveri P., and Chavan V., "Load Balancing on Cloud Data Centers," *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 3, no. 1, pp.1-4, 2013.

[18] Subashini S. and Kavitha V., "A Survey on Security Issues in Service Delivery Models of Cloud Computing," *Journal of Network and Computer Applications*, vol. 34, no. 1, pp. 1-11, 2011.

[19] TaheriMonfared A. and Jaatun M., "Handling Compromised Components in an IaaS Cloud Installation," *Journal of Cloud Computing*, vol. 1, no. 16, 2012.

[20] Vouk M., "Cloud Computing Issues, Research and Implementation," *in Proceeding of 30th International Conference on Information Technology Interfaces*, Croatia, pp. 31-40, 2008.

[21] Wickremasinghe B., Calirous R., and Buyya R., "A Cloudsim-based Visual Modeler for Analyzing Cloud Computing Environment and Application," *in Proceeding of 24th IEEE conference on Advanced Information Networking and Applications*, Washington, pp. 446-452, 2010.

[22] Wu L., Garg S., and Buyya R., "SLA-Based Resource Allocation for SaaS in Cloud Computing Environments," *in Proceeding of 11th IEEE/ACM International Symposium on Cluster*, Washington, pp. 195-204, 2011.

[23] Wu M. and Sun X., "Grid Harvest Service: A Performance of Grid Computing," *Journal of Parallel and Distributed Computing*, vol. 66, no. 10, pp. 1322-1337, 2006.

[24] Yeo C. and Buyya R., "Service Level Agreement based Allocation of Cloud Resources: Handling Penalty to Enhance Utility," *in Proceeding of International Conference on Cluster Computing*, Bostan, 2005.

Radhakrishnan Ayyapazham received his M.E. degree in Computer Science and Engineering from Anna University Chennai, India. He is currently persuing PhD degree and working as Assistant Professor, Department of Computer Science and Engineering at Anna University, Tirunelveli Region, India. He has 12 years of academic experience. He participated many national and international confernces. His research interest on cloud computing focusing on Infrastructure as a Service.

Kavitha Velautham received her B.E degree in Computer Science and Engineering in 1996 from M.S. University and M.E degree in Computer Science and Engineering in 2000 from Madurai Kamaraj University. Presently she is working as Associate Professor in the Department of CSE at University College of Engineering Nagercoil, Anna University, India. Currently, under her guidance 11 Research Scholars are pursuing Ph.D as full time and part time. Her research interests are Wireless networks, Mobile Computing, Network Security, Wireless Sensor Networks, Image Processing, Cloud Computing. She has published many papers in Conference, National and International Journal in areas such as Network security, Mobile Computing, wireless network security, and Cloud Computing.