Analyzing Fuzzy Models for VM Resource Usage Prediction in Virtualized Data Center

Mya Sithu
University of Computer Studies, Yangon
ms.myasithu@gmail.com

Abstract

Data Centre (DC) administrators try to delivering performance guarantees while managing resources for utilization in terms of reducing cost. With the advent of server consolidation provided by virtualization technology, multiple heterogeneous virtual machines (VMs) can be coexisted on a physical server and shared resources together. Fixed allocation of resources to VMs is not the optimal allocation method as over provisioning and under provisioning can be caused. For dynamic allocation, simultaneous on demand provisioning of shared physical resources to VMs becomes the key challenge.

This paper proposes resource usage prediction system by making analysis on the accuracy of three different models; Fuzzy modeling, adaptive Fuzzy modeling and Neuro-Fuzzy modeling. To evaluate the efficiency of three different models, experiments are carried out by workload-resource mapping and resource-resource mapping approaches. CPU profiles from real world data centre are used to analyze through a simulating program. Experimental results show that the proposed resource prediction models can predict well next time interval resource usage of virtual machine even in the condition of unexpected high spikes CPU.

Keywords- resource prediciton; virtualized data center; resource provisioning; fuzzy, aadaptive fuzzy; neuro-fuzzy

1. Introduction

Virtualization is gaining popularity in enterprise environments as a software-based solution for building shared hardware infrastructures. Forrester Research estimates that businesses generally end up using somewhere between 8 and 20 percent of the server capacity they have purchased. Virtualization technology helps to achieve greater system utilization while lowering total cost of ownership and responding more effectively to changing business conditions. For large enterprises, virtualization offers an ideal solution for server and application consolidation in an on-demand utility [6].

In Virtualized Data Centre (VDC), resource allocation plays an important role. Applications served by a data center are usually business critical applications with Quality-of-Service (QoS) requirements. The resource allocation needs to not only guarantee that a virtual container always has enough resources to meet its application's performance goals, but also prevent over provisioning in order to reduce cost and allow the concurrent hosting of more applications [5]. Moreover, fixed allocation is the cause of over provisioning and under provisioning as applications in data centre has time varying workload mixes. In order to figure out these provisioning problems and avoiding Service Level Agreements (SLA) deviations, dynamic resource allocation system becomes a key challenging problem.

In an efficient dynamic allocation system, real time monitoring of VM resource usages and allocation of underlying physical host resources to VM are needed to perform in time. In order to perform such difficult tasks, learning of previous VM resource usage nature is becoming important to make resource allocation decisions. Different VMs can run different kinds of applications which have different workload patterns. By learning the relationship between workload and its CPU usage, next time interval resource usage can be predicted based on current time monitored resource utilization. QoS of applications, targeted response time of data centre applications should be taken into account as SLA deviations are one of the main issue in VDC.

In this paper, resource usage prediction of VMs is presented intending to be supportive in dynamic resource allocation in VDC. Previous researches on this topic make predictions by time series prediction and work flow models [3, 4, and 5]. But in our proposed models, cluster based Fuzzy modeling; adaptive Fuzzy modeling and Neuro-fuzzy modeling are chosen. These methods are analyzed and evaluated through two approaches; workload-resource mapping and resource-resource mapping. Workloads and CPU resource usages from five different test beds with heterogeneous hardware specifications are used for experiments with three methods. Experiments are carried out a simulation program in Matlab.

The rest of the paper is structured as follows. Section 2 presents the related work of this paper. Section 3 describes resource prediction models in a virtualized data center. Analysis on prediction models are explained in Section 4. Discussions of experimental results on prediction models are described in Section 5. It is followed by conclusion and future work in section 6.

2. Related Work

Some of the previous works of researchers that are related to the proposed system are described in this section. Sandpiper [10] automates the task of monitoring and detecting hotspots, determining a new mapping of physical to virtual resources and initiating the necessary migrations. But Sandpiper did not take into account the complicated and uncertain relationship between the system's parameters. P.Paddala [9] presents their system as automated control of multiple virtualized resources. Auto control is a resource control system that automatically adapts to dynamic workload changes to achieve application service level objectives (SLO). The model estimator captures the complex relationship between application performance and resource allocations, while the MIMO controller allocates the right amount of multiple virtualized resources to achieve application SLOs. In online model estimator, they used adaptive modeling approach to capture the complex behavior of enterprise applications.

Jing Xu [6] proposed a two level resource management system to dynamically allocate resources to individual virtual containers. It uses local controllers at the virtual container level and a global controller at the resource pool level. The local controller is based on fuzzy modeling approach and fuzzy prediction to deal with the complexity and uncertainties of dynamically changing workloads and the global controller is based on the profit model to maximize the data center's profit. They used fuzzy logic and no tuning method is considered. In [11], VMware's Distributed Resource Scheduler solves the CPU and memory pressure by performing load balancing dynamically. But VMware's DRS cannot utilize application logs to have better placement decisions. Moreover, DRS is only efficient for homogeneous virtualized environment.

In [3], H.Choi proposed a flow based prediction scheme which forecasts future resource usage by using two-layered neural network. They compared their result with auto regression prediction scheme in terms of accuracy and overheads. They developed monitoring tool and migration tool for their flow based scheme. Although it is concluded that the flow-based prediction scheme outperforms AR scheme, the prediction is totally impossible without the pattern on data.

3. Resource Prediction Models

VM resource usage predictions are the most important feature of auto dynamic resource allocation systems. In VDC, real time monitoring of the resource usages of VMs and dynamic allocation of physical machine resources are needed to perform in time before overloading and underloading occurs. Although popular hypervisors can give live migration feature, it can cost additional overheads to the loaded machine. Among the various prediction models, rule based Fuzzy inference system is chosen in this paper because of its ability of handling uncertainties and linguistic values. In our proposed system, three different kinds of Fuzzy models are analyzed via two approaches; workload-resource mapping and resource-resource mapping. The former is applicable when the workload can be monitored and characterized. Actually, a single VM can host a variety of applications and there is no standard way of measuring the workloads of them. The second approach resource-resource mapping requires information about resource usage which is easy to obtain by monitoring system parameters. In both approaches, Fuzzy modeling, adaptive Fuzzy modeling and Neuro-Fuzzy modeling are used to predict resource usages.

3.1 Fuzzy Modeling

Fuzzy logic [1, 2, 13] is a tool to deal with uncertain, imprecise, or qualitative decision-making problems. Unlike Boolean logic, where an element x either belongs or does not belong to a set A, in fuzzy logic the membership of x in a fuzzy set F has a degree value (called fuzzy value) in a continuous interval between 0 and 1 representing the extent to which x belongs to F. Fuzzy sets are defined by membership functions that map set elements into the interval [0, 1]. One of the most important applications of fuzzy logic is the design of fuzzy rule-based systems. These systems use “IF-THEN” rules (also called fuzzy rules) whose antecedents and consequents use fuzzy-logic statements to represent the knowledge or control strategies of the system. The collection of fuzzy rules is called a rule base.

The process of formulating the mapping from inputs to outputs using fuzzy rules is called the fuzzy inference (FIS) mechanism. Since fuzzy rules use fuzzy sets and their associated membership functions to describe system variables, fuzzification and defuzzification functions are necessary for translating between numeric values and fuzzy values.

In the first step of proposed Fuzzy modeling, input and output data pairs in the data set are clustered using subtractive clustering. After projecting the clusters onto the input space, the antecedent parts of the fuzzy rules can be found. The consequent parts of the rules can then be simple functions. In this way, one cluster corresponds to one rule of the Takagi-Sugeno-Kang (TSK) model. The TSK model is composed of the IF-THEN rules of the following form.

$$R_{(r)}: if\ x_1\ is\ A_r^1\ and\ A_r^2\ and\ \dots and\ x_m\ is\ A_r^m$$

$$then\ y_r\ is\ fr(x) \tag{1}$$

$$where\ f_r(x) = \alpha_r^0 + \alpha_r^1 x_1 + \cdots + \alpha_r^m x_m \tag{2}$$

in which $(r_m = 1, \dots, n)$ and $(x_j (1 \le j \le m)$ are the input variables, y_r is the output variable, A_r^m are fuzzy sets, and $f_r(x)$ is a linear function. Max-Min function and centroid method is used for fuzzification and defuzzification respectively. Among the various types of membership functions, Gaussian membership functions are chosen for this model.

The fuzzy model’s parameters determined by the calculated centers and ranges of data clusters are also stored in a database. Once the fuzzy model is built up, the fuzzy inference functions periodically process the fuzzy rules kept in the knowledge base to determine the resource needs based on the currently monitored workload and required QoS. Figure 1 shows the proposed resource prediction system using Fuzzy Modeling.

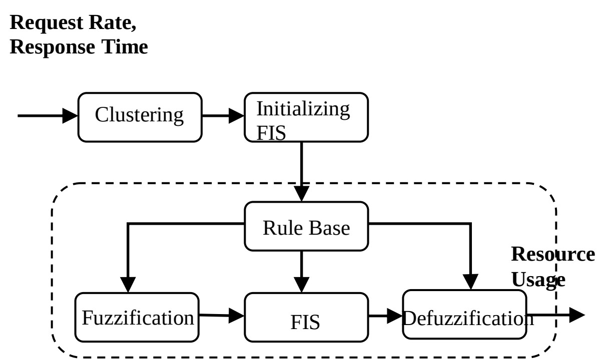

Figure 1. Fuzzy Modeling

3.2 Adaptive Fuzzy Modeling

Fuzzy modeling can only consider offline model learning and it is not convenient when the nature of learned workload patterns or system conditions change. To overcome this condition, the proposed adaptive Fuzzy modeling considers updating the rule base repetitively based on currently monitored workload and resource usages. In this method, rules are added to FIS as soon as the new information is learnt so that up-to-date clusters can always be provided for prediction. In this method, 11 membership functions; region1 (R1) to region 11 (R11) are chosen for both of the approaches of workload-resource mapping and resource-resource mapping as shown in Figure 2.

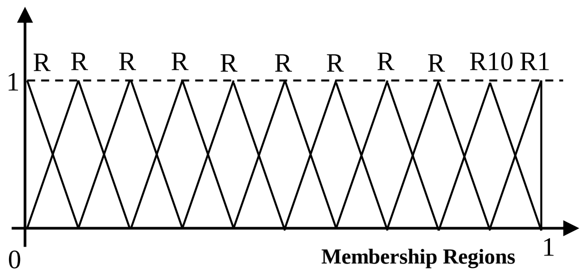

Figure 2. 11 Membership Functions

3.3 Neuro-Fuzzy Modeling

Neural nets [1,8] use a number of simple computational units called "neurons". First, the propagation function combines all inputs X_i that stem from the sending neurons. The means of combination is a weighted sum, where the weights w_i represent the synaptic strength. To express a background activation level of the neuron, an offset (bias) is added to the weighted sum. The activation function computes the output signal Y of the neuron from the activation level. The information enters the neural net at the input layer. All layers of the neural net process these signals through the net until they reach the output layer. The objective of a neural net is to process the information in a way that is previously trained. For training, neural nets use "learning algorithms". The learning algorithm modifies the individual neurons of the net and the weight of their connections [1].

The Neuro-fuzzy controller uses the neural network learning techniques to tune the membership functions while keeping the semantics of the fuzzy logic controller intact. Neural networks offer the possibility of solving the problem of tuning. Although a neural network is able to learn from the given data, the trained neural network is generally understood as a black box. Neither it is possible to extract structural information from the trained neural network nor can we integrate special information into the neural network in order to simplify the learning procedure. On the other hand, a fuzzy logic controller is designed to work with the structured knowledge in the form of rules and nearly everything in the fuzzy system remains highly transparent and easily interpretable. A combination of neural network and fuzzy logic offers the possibility of solving tuning problems and design difficulties of fuzzy logic [8]. The resulting network will be more transparent and can be easily recognized in the form of fuzzy logic control rules or semantics [7].

3.3.1 Adaptive Neuro-Fuzzy Inference System (ANFIS)

As for the third kind of experiment in our proposed system, ANFIS simulation model in Matlab is used. Using a given input/output data set, ANFIS constructs a fuzzy inference system (FIS) whose membership function parameters are tuned using either a backpropagation algorithm alone or in combination with a least squares type of method. This adjustment allows the fuzzy systems to learn from the data which are used for modeling.

The parameters associated with the membership functions changes through the learning process. The computation of these parameters (or their adjustment) is facilitated by a gradient vector. This gradient vector provides a measure of how well the fuzzy inference system is modeling the input/output data for a given set of parameters. When the gradient vector is obtained, any of several optimization routines can be applied in order to adjust the parameters to reduce some error measure. This error measure is usually defined by the sum of the squared difference between actual and desired outputs. ANFIS uses either back propagation or a combination of least squares estimation and backpropagation for membership function parameter estimation.

4. Analysis on Prediction Models

For analysis of three different Fuzzy models, experiments are performed via workload-resource mapping and resource-resource mapping approaches. In the first experiment of workload-resource mapping, user request rate and targeted response time of application owners are taken as input and VM's CPU resource usage is produced as output. After completing the first kind of experiment, second experiment is dug up taking only the user request rate as input and producing CPU resource usage of VM.

4.1 Workload-Resource Mapping (W-R mapping)

To analyze the relationship between workload (web user access rate in this paper) and CPU resource usage, the experimented data shown in Figure 3 are used [12]. This experimented data are collected from the test bed which is shown in Table 1.

Table 1. Test Bed 1

Machine	Two Dell OptiPlex 755
CPU	2.33 GHZ Intel Core
RAM	3 G
Hard Disk	160 GB
Ethernet	100 M

It can be seen from the figure that the relationship between workloads and CPU usage is learnt through experiments in which CPU allocation is varied from 10% to 90% with increment of 5% based on 100% total resource usage. At each resource allocation, the request rate of the application is changed from 100 to 580 per second with the increment of 50. It can be shown by the figure that the response time of the server can be reduced by allocating more CPU percentage to it.

However, when CPU allocated to the driver domain is fewer than 15% of the total resource, it becomes the bottleneck of the system because of the increased response time. Therefore, it is needed to take into account that resources of physical host between 15% and 85% should be assigned to virtual machines.

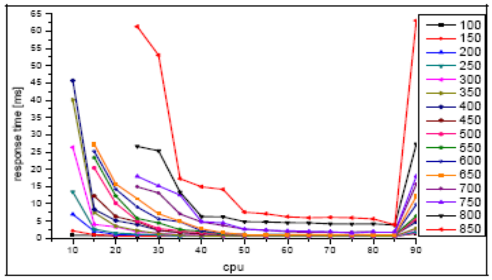

Figure 3. Relationship between request rate, response time and CPU%

4.1.1 Fuzzy Modeling for W-R Mapping

In the first experiment of fuzzy-modeling for workload-resource mapping, three dimensional data sets (user request rate and required QoS of application owner (response time) are as input and CPU usage as output) are used. In this kind of dataset, rules are created from the input, output pairs where antecedent part contains two parameters and the consequent part of each rule has one parameter. These three dimension data are firstly clustered by subtractive clustering to generate initialized Fuzzy Inference System (FIS). FIS consists of rules where each rule is representing each cluster.

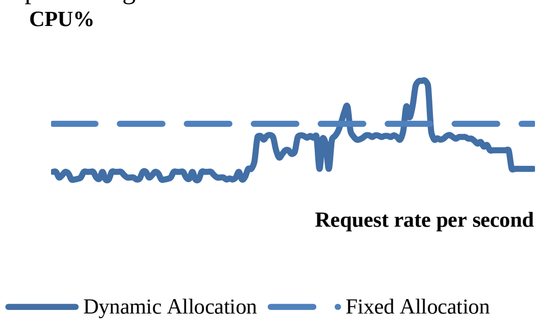

Figure 4. Dynamic and fixed allocation

In the experiment shown in Figure 4, the targeted response time is regarded as 6 ms. If fixed allocation is chosen for this case, VM's CPU allocation of 35 percent must be used to get preferred response time in terms of worst case provisioning. But it can be seen that resource allocation can be greatly reduced because of dynamic allocation.

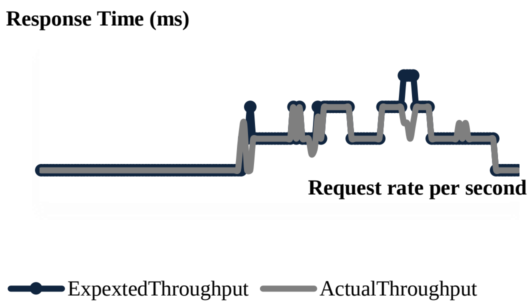

Figure 5. Response time of dynamic and fixed allocation

When response time of dynamic allocation is compared with its kind of fixed allocation, it can be seen that there is not much difference between them. On the other hand, it can be said that no SLA deviation can occur from dynamic resource prediction system as shown in Figure 5. Actual and predicted CPU resource usages are shown in comparisons in Figure 6. In this figure, the first part of the label of X-axis indicated that user request rate per second and the second part of it represents the response time per millisecond.

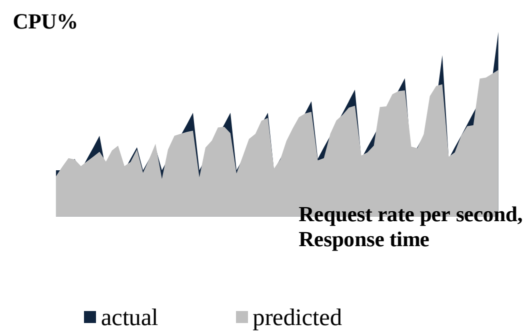

Figure 6. CPU usage prediction of three-D data sets

In the second experiment of fuzzy modeling for W-R mapping, two dimensional data set (user request rate is taken as input and CPU resource usages as output) is used. Same test bed data pairs shown in Table 1 are used for this kind of experiment and it is found that better prediction accuracy can be achieved in this manner. Figure 7 shows the difference of prediction accuracy for this kind of experiment.

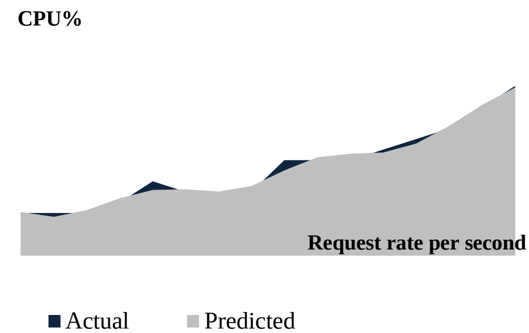

Figure 7. CPU usage prediction of two-D data sets

From the experimental results getting from Fuzzy modeling as shown in Figure 6 and 7, it is found that Fuzzy modeling with two-D data sets can give better accuracy than three-D data sets. The reason is that data pairs are difficult to make cluster regions in the former one, as two dimensions of user request rate and response time data pairs have very different values in ranges and it makes less accuracy of prediction.

4.1.2 Adaptive Fuzzy Modeling for W-R Mapping

The similar data are used again to make the experiment using adaptive fuzzy prediction. Instead of clustering based on the data range, in this method, a single initial rule (workload, response time as input and CPU as output) is stored in the rule base. As the prediction results are based on the rules stored in the rule base, adaptive fuzzy has low prediction accuracy in its earlier stages. But it stores the new rule at the same time of its prediction. After updating the rule base with collected workload and CPU with the required performance of service level agreements at next time intervals, it is found the adaptive fuzzy model has less difference between actual and predicted resource usages. Figure 8 shows the actual and predicted CPU usage of before updating the rule base and after updating the rule base by Adaptive Fuzzy modeling for the first kind of dataset.

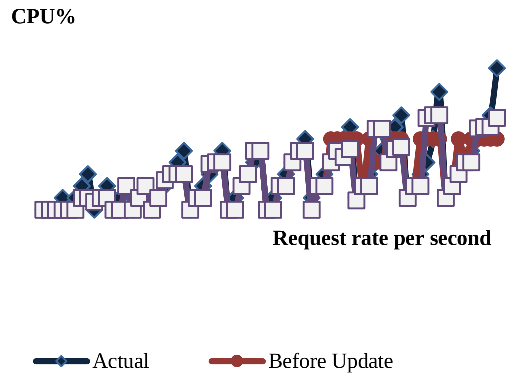

Figure 8. CPU% prediction of three-D data sets in adaptive Fuzzy modeling

In Figure 8, it can be seen that there is less accuracy in prediction between predicted and actual CPU resource usage before finishing the rule base constructing. But after N times interval, rule base is already constructed with enough number of rules for predicting the resource usage based on current monitored workload. At this time, the prediction accuracy is higher and there is not much difference between the actual and predicted value.

Figure 9 shows the predicted and actual usage of two dimensional data sets, user request rate is taken as input and CPU usage is taken as output.

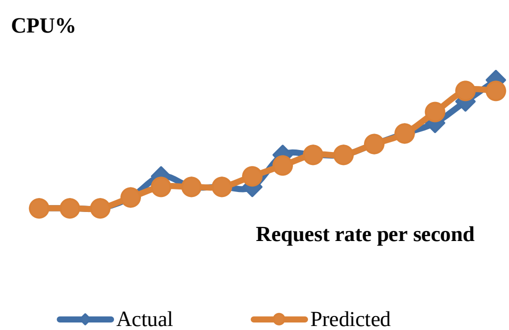

Figure 9. CPU% prediction of two-D data sets in adaptive Fuzzy modeling

Although the Fuzzy modeling has much difference in three dimensional data sets and two dimensional data sets, adaptive Fuzzy modeling has not so much difference in them. The only factor which can cause difference in this method is the condition of before and after updating the rule base for reasonable number of rules.

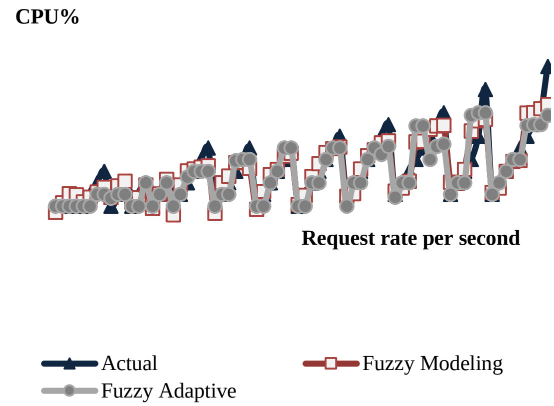

Figure 10. Comparisons of three-D data sets

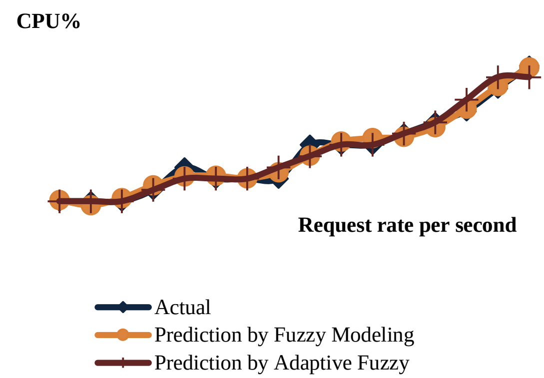

Figure 11. Comparisons of two-D data sets

Figure 10 and Figure 11 show the difference of predicted CPU usage and actual CPU usage of fuzzy modeling and adaptive fuzzy modeling .Two data sets are used and both of the data set contains 100 records in which 50 records are used for training purpose and other 50 records are used for testing. The difference is that the first one has repetitive and stable workload nature and the second one has unexpected pattern in its testing records. By comparing two methods for workload-resource mapping, it is found that the adaptive method can predict better unprecedented CPU spikes. But cluster based fuzzy model system is more optimal if the data center server have very repetitive pattern as it can make better prediction. In a few words, it can be said that Fuzzy modeling is better for offline modeling whereas adaptive Fuzzy prediction is better for online modeling.

4.1.3 Neuro-Fuzzy Modeling for W-R Mapping

Table 2 shows the training error and testing error produced by ANFIS model when experimenting with the same data sets used for previous methods. Firstly, data pairs of three dimensional data are taken for training. Every 100 records of data set are used for training and all of the 150 records are used again for testing purpose. ANFIS has two methods for generating FIS whereas the first generating method use grid partitioning (genfis1) and the second method use subtractive clustering (genfis2). In this section, experiments are carried out using both of genfis1 and genfis2 but it is found that differing generating methods cannot take much effect on prediction accuracy.

Table 2. Results of genfis1 and genfis2

Prediction Errors				
FIS	**Grid Partitioning (Genfis1)**		**Subtractive Clustering (Genfis2)**	
Data Set	Three -D data	Two - D data	Three- D data	Two-D data
Training Error	10.3762	2.03485	10.4602	2.0242
Testing Error	10.3762	2.0349	10.4602	2.0251

In every experiment of ANFIS simulation model with different data sets, 100 numbers of epochs are chosen for training and tuning the membership functions. 100 epochs means 100 times running of backpropagation algorithm to make less difference between actual and predicted values. In Table 2, experimental results of ANFIS Fuzzy models with different data sets, different generating methods are shown.

4.1.4 Discussions on Experimental Results of Prediction Models

Comparisons of prediction errors of three proposed models are shown in Table 3. Three dimensional data sets contains two input and one output pairs (User request rate, targeted response time and CPU usage) where two dimensional data contains one input and one output pairs (User request rate and CPU usage).

Table 3. Average errors

Test Bed	Method	Data Sets	Average Error
Test Bed1	Fuzzy Modeling	Three dimensional	3
		Two dimensional	≅ 0
	Adaptive Fuzzy Modeling	Three dimensional	3.5
		Two dimensional	1.86
	Neuro-Fuzzy Modeling	Three dimensional	10.4
		Two dimensional	2

4.2 Resource-Resource Mapping (R-R mapping)

W-R mapping is only applicable in the data center environment when the application workload can be characterized and monitored. However, data centers can host a variety of applications which are very different from each other so that there is no standard way of measuring workloads. In some cases it is hard to describe an application workload using a few metrics like request rate. The second proposed approach resource-resource mapping only requires information about the resource usage, which is easy to obtain by monitoring system-level metrics. The basic idea is to determine future resource needs on the basis of observations of past resource usage using fuzzy system. In this approach, first three intervals of resource usage is taken as input and the next two time intervals of resource usage is predicted based on the first three intervals as shown in Figure 12.

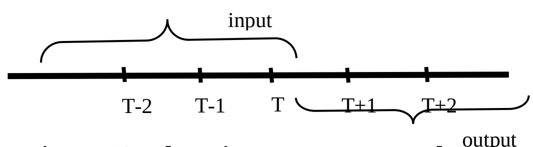

Figure 12. Three input- two output data for Fuzzy Rule Construction

The resource usage profiles from the four test beds which is shown in Table 4 are collected as data sets for all kinds of experiments. Test Bed 2 and Test Bed 3 contain eight VMs. Test Bed 4 and Test Bed 5 contain physical machines and no virtual machine is created in those test beds. But they are also chosen for our experiments because different kinds of data sets are needed to use to check prediction efficiency.

Table 4. Test Bed 2, 3, 4 and 5

Test Bed	Test Bed 2	Test Bed 3	Test Bed 4	Test Bed 5
PM	Quad Core 3 Hz Processor	A Pair of Quadcore Processor	CPU Intel 4 x 2.7 GHz Dual Core	CPU Intel 4 x 2.7 GHz Dual Core
	128 GB of RAM	IBM Blade Centre HS22	7 GB of RAM	4G of RAM
VM	A Quad Core CPU	A Quad Core CPU	-	-
	16 GB of RAM	16 GB of RAM	-	-

In test bed 2 and 3, VMs are assigned as exchange server, oracle server, web server and scan/read server. Different resource usage profiles on these VMs are used for experiments. Physical machine in test bed 4 has Window Server 2003 enterprise edition and it runs web server and JBOSS event server. It is connected to database server which has different hardware specifications. But in our proposed system, CPU profiles from web server machine are used as input. Test bed 5 is also the web server which has between 500 and 600 daily average user.

4.2.1 Fuzzy Modeling for R-R Mapping

In the experiment using fuzzy modeling, data collected from four test beds are used. After carrying out the experiments with Fuzzy modeling in the same way doing in the W-R mapping, it is found that this kind of experiment has best accuracy for its clustered data because it can give 100 percent accuracy rate for clustered data pairs. Figure 12 shows the rules created for VM1 in test bed 2. In this figure, it is found that 22 rules are created in the region of 10 and 20. That means the data set which is trained for this FIS has CPU usages between 10 percent and 20 percent. Clustered and rules are formed within these ranges and they are shown in the rule view of Figure 13.

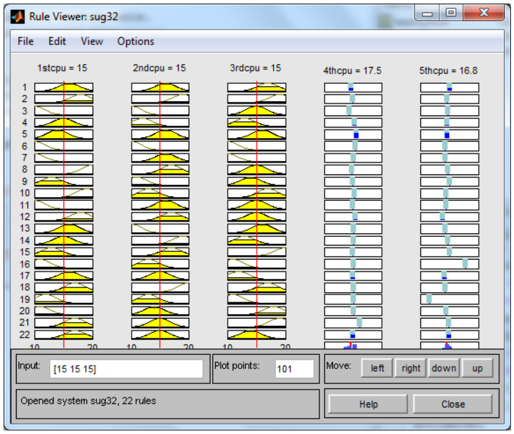

Figure 13. Ruleview of VM1, Test Bed 2

4.2.2 Adaptive Fuzzy Modeling for R-R Mapping

In the same way of W-R mapping, there is no rule info the rule base at first. After the first m + n measurements are obtained, the first rule is generated and stored in the rule base. Afterwards, at each sampling point, a rule is constructed and the rule base is updated. This updating procedure makes the proposed fuzzy prediction capable of self-learning the resource usage behavior of the managed virtual container. Rules can be conflicted if they have same IF part and different THEN parts. When updating the rule base a reliability index is computed for each rule as J_i = the number of occurrences of rule i. Whenever a rule is generated, the system scans all the rules stored in the rule base. If there is a matching rule (i.e., a rule in the same domain), the value of J is increased by 1. Otherwise, the new rule is added to the rule base and J is initialized to 1. If there are existing conflicting rules, which one takes effect is determined by the value of the reliability index. The rule with the highest reliability index is activated, indicating that the active fuzzy rule appears more frequently than the other conflicting rules. If conflicting rules have the same value of reliability index, the one that appeared most recently is activated. Figure 14 shows the prediction accuracy for CPU usages of **VM** 1 from Test Bed 2.

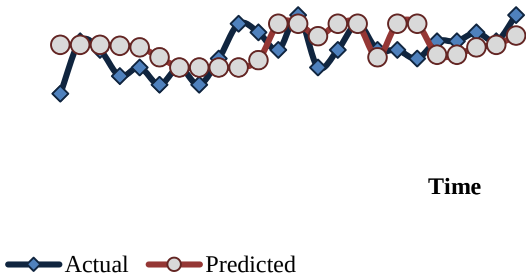

Figure 14. CPU% prediction for VM1, Test Bed 2

4.2.3 Neuro-Fuzzy Modeling for R-R Mapping

At this time, ANFIS is used for the experiments of resource-resource mapping. Experiments are made using both of genfis1 and genfis2. Table 5 shows the comparison of root mean squared error produced by genfis1 and genfis2 of resource usage prediction on eight VMs on Test Bed 2. At each time of experiments, epoch 100 is used as training times.

Figure 15 shows the comparison of real CPU usage and predicted CPU usage of ANFIS when predicting the CPU usage of VM1 on Test Bed 2 by grid partitioning (genfis1).

Table 5. Average Errors of genfis 1 and 2

VMs of Test Bed 2	genfis1		genfis2	
	Training Error	Testing Error	Training Error	Testing Error
VM1	0.0154	0.0130	0.0239	0.0200
VM2	0.0146	0.0120	0.0092	0.0052
VM3	0.0100	0.0080	0.0140	0.0101
VM4	0.0118	0.0090	0.0173	0.0150
VM5	0.0059	0.0035	0.0086	0.0045
VM6	0.0110	0.0096	0.0141	0.0139
VM7	0.0190	0.0152	0.0052	0.0042
VM8	0.0087	0.0037	0.0154	0.0140
Average Error	0.0120	0.0092	0.0134	0.0108

In the predicted CPU usage sub-figure, the single line of predicted CPU usage (green line) can be seen as it is identical to the real CPU usage. Error curves and ANFIS prediction errors are also shown as sub-figures. Figure 16 shows the results of subtractive clustering (genfis2).

Figure 15. Experimental Results of genfis1

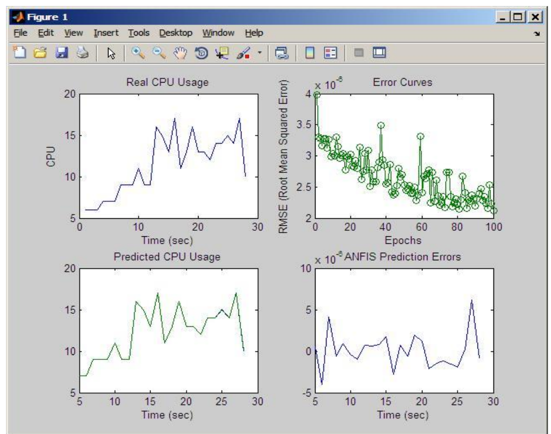

Figure 16. Experimental Results of genfis 2

4.3 Discussion on Experimental Results

After experiments are carried out, it can be concluded that Fuzzy modeling and Neuro-Fuzzy is more suitable for stable and periodic pattern while adaptive Fuzzy modeling has the ability of handling unexpected spikes of the pattern. But Neuro-Fuzzy needs time and it is rising as if the more number of epochs are choosing for training phase. In a few words, it can be concluded that Fuzzy modeling is best for workload-resource mapping and adaptive Fuzzy and Neuro-fuzzy is more suitable for Resource-Resource mapping. Table 6 shows the comparisons of three methods resulting from the experiments.

Table 6. Comparisons of Fuzzy models

Test Bed	Method	Avg Err	Advantages	Disadvantages
TB 1, TB 2, TB 3, TB 4, TB 5	Fuzzy Modeling	$\cong 0$	-Suitable for Periodic Pattern -Best Accuracy for Periodic Pattern	-Worst Case Prediction with unexpected data pattern
	Adaptive Fuzzy Modeling	1.52	-Suitable for more dynamic Usage Pattern -Able to handle data pattern with high peak load	-Higher absolute error while updating rulebase
	Neuro-Fuzzy Modeling	0.127	-Adjustable membership functions	-Take considerable time for training

5. Conclusion

This paper presents the fuzzy-based virtual machine resource prediction system for VMs in VDC. To enable to accurately estimate the resource demands of VMs, two approaches; W-R mapping and R-R mapping are proposed. In both of two approaches, predictions by three models, Fuzzy modeling, adaptive Fuzzy modeling and Neuro-Fuzzy models are analyzed. According to our experimental results, we are sure that fuzzy based logic decision making has the capability of controlling the nonlinear relations, uncertainty and dynamic nature. It is also able to perform well in situations with unpredictable and abrupt peak loads. Fuzzy modeling outperforms for offline data. Adaptive Fuzzy modeling is more suitable for online data monitoring and rule base updating. Neuro-Fuzzy has efficient membership functions and it stands middle of the other two methods although much training time is needed

References

[1] B.Kosko. Neural Networks and Fuzzy Systems. First Edition, Prentice Hall of India, 1997.

[2] D.Nauck, F.Klawonn, and R.Kruse, Combining neural networks and fuzzy controllers, Fuzzy Logic in Artificial Intelligence, Springer, 1993, pp. 35-46.

[3] H.Choi, K.Lee and S.ParkA. Flow-Based Prediction Scheme to Manage Resources in Enterprise Data Centers. In Proc. IEEE, July 2008, pp. 285-290.

[4] H.Won Choi, H.Kwak, Autonomous Learning for Efficient Resource Utilization of Dynamic VM Migration, In Proc. 22nd annual international conference on Supercomputing, ISC'08.

[5] J.Dejun, G.Pierre and C.H.Chi, Autonomous Resource Provisioning for Multi Service Web Applications, WWW 2010, April 26–30, 2010, Raleigh, North Carolina, USA. 2004 ACM.

[6] J.Xu, M.Zhao, J.Fortes, R.Carpenter and M.Yousif. Autonomic Resource Management in virtualized data centers using fuzzy logic-based approaches. In Proc. IEEE, July 2008.

[7] K.Begnum and M.Disney. Decision Support for Virtual Machine Re-Provisioning in Production Environments. 21st Large Installation System Administration Conference, LISA' 07.

[8] P.J.Werbos, "Neuro control and fuzzy logic: connections and design," Int. J. Approximate Reasoning, Vol. 6, Feb. 1992, pp. 185-220.

[9] P.Padala, X.Zhu, M.Uysal and Z.Wang. Automated Control of Multiple Virtualized Resources. In Proc. EuroSys'09 ACM, April 2009.

[10] T.Wood, P. Shenoy,A. Venkataramai, and M. Yousif. Black Box and Gray Box Strategies for Virtual Machine Migration . Computer Networks Journal Special Issue on Virtualized Data Center 2009.

[11] http : // www. vmware. com/ pdf / server consolidation.pdf.

[12] Y.Zhang, A.Vestabros, Friendly Virtual Machines, VEE'05, June 11-12, 2005, Chicago, Illinois, USA. 2005 ACM.

[13] Zadeh, L.A.: Fuzzy Sets. Inf.Control 8(3), pp. 338-353 (1965).