International Journal of Grid and Distributed Computing
Vol.7, No.2 (2014), pp.41-52
http://dx.doi.org/10.14257/ ijgdc.2014.7.2.05

The Method of Predicting Average Response Time of Cloud Service Based on MGM (1, N) - BP Neural Network

Jun Guo, Qun Ma, Qingmin Ma, Yongming Yan and Qingliang Han

Northeastern University; Software Testing Center of Shandong Province
boatheader@163.com;maqun159@yeah.com

Abstract

In the cloud computing environment, according to the predicting average response time of service, it can adjust to the follow-up system, so that the response time of the system is acceptable. The traditional methods of predicting average response time of serve mainly include the method of gray predicting and neural network model, but the two methods face several problems, such as longer processing time and unsuitable to larger volatility data. According to the above problems, the paper proposes the method of predicting average response time of cloud service based on the MGM (1, N) - BP neural network, the combination of two methods of predicting can use less sample information, it can get a high precision of predicting result and it can also predict the volatile system. Experimental results show the feasibility and effectiveness of the method.

Keywords: *MGM(1,N)-BP neural network; cloud service; the average response time; the predicting method*

1. Introduction

In a time interval, the average response time is the mean value of virtual machine service response time [1]. In cloud computing environment, according to the predicted average response time of service, it can timely adjust to the follow-up system, so that the response time of the system is acceptable [2].

The traditional methods of predicting average response time of serve mainly include the method of gray predicting and neural network model. The former can find out the rule of every period in the case of less data and establish the load forecasting model. The grey forecasting model method is simple, with less data volume, but not suitable for large volatility data [3]. The latter is a multi-level network linked by several basic units according to some rules. Employing the network structure to forecast data will lead to long training time and be easily in local state [4].

According to these problems, the paper proposes the method of predicting average response time of cloud service based on the MGM (1, N) - BP neural network that combines the MGM(1,N) [5] predicting method and BP neural network predicting method [6]. The method can use less sample information, get highly precise results and predict the volatile system. Experimental results show the feasibility and effectiveness of the method.

2. The Predicted Process

The predicted process of average response time is mainly divided into four stages, with a virtual machine instance hosting service, for example, predicting the basic process of response time is shown in Figure 2.1. The average response time of the

ISSN: 2005-4262 IJGDC

service on all virtual machine instances is mean value of all virtual machine instance response time.

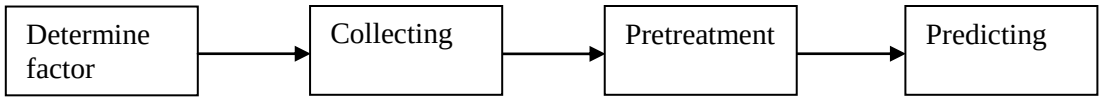


Figure 2.1. The Predicted Process

(1)Determine factor: the factors of effecting virtual machine service mainly have: CPU, memory footprint, disk usage and network bandwidth utilization rate and so on. This paper will adopt correlating analysis method to determine the close degree between the average response time and the influencing factors, choosing the factor with larger correlation degree as the independent variable of the subsequent predicting model. The calculation steps are:

- According to the correlation $\Delta_i^{(k)} = | x_0^{(k)} - x_i^{(k)} |$, calculating $\varepsilon_i^{(k)}$:

$$\varepsilon_i^{(k)} = \frac{min\ min\{\Delta_i^{(k)}\} + \rho max\ max\{\Delta_i^{(k)}\}}{\Delta_i^{(k)} + \rho max\ max\{\Delta_i^{(k)}\}} \tag{2.1}$$

Among it, $\Delta_i^{(k)}$ is the result that data sequence of candidate index factor minuses average response time of service. ρ is resolution function . The value space is [0,1]. Usually, ρ is 0.5.

- Calculating correlation, the formula is:

$$\gamma_i = \frac{1}{n}\sum_{k=1}^{n}\varepsilon_i^{(k)} \tag{2.2}$$

(2) Collection data: before the starting of the prediction process, it should record datum of CPU utilization, memory footprint, disk usage and network bandwidth.

(3) Pretreatment: on virtual machine, history datum of impacting factor of the service average response time may exist in different dimensions and big data, which can affect the prediction accuracy of the average response time. So it needs pretreatment. Pretreatment has two types. The first one is processing, the next one is data standardization. Pretreatment may remove the influence of different data source. The data processing is used to find and solve uncertain data. The data standardization has two types: data with chemokine and dimensionless processing. The first one may dispose different kinds of data. If the process do not data with chemokine, the result should be wrong. So data with chemokine is significant. The next one may solve the problem about data comparability. Changes of memory and CPU are shown in Figure 2.2 and Figure 2.3.

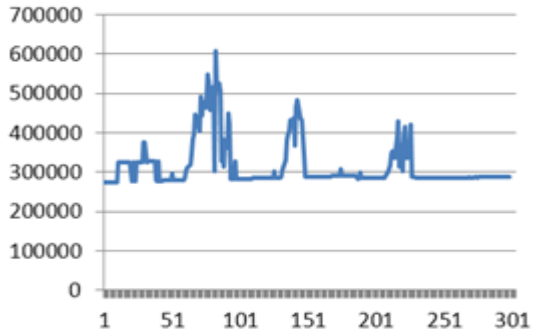


Figure 2.2. The Usage of Memory of the Service in VM

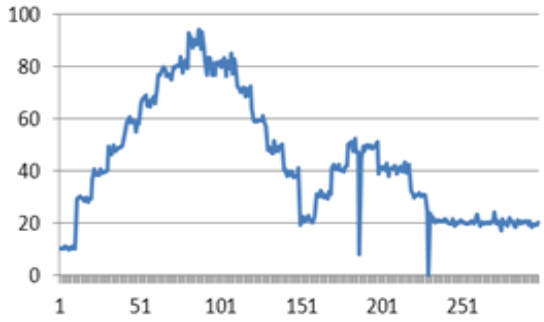


Figure 2.3. The Usage of CPU of the Service in VM

It can be seen from figure, more service need memory, bigger is changes of memory. Change of memory is between 280000KB and 620000KB. Change of CPU is between 15% and 95%. But it is very difficult for predicting service average response time, dimension of memory and CPU is not consistent. Standardized treatment is needed in the system. Max and min of index data are unknown. The historical data of service indicators may be performed by using Z-score standardized method.

Dealing with data before the system begins is very important. It can input accurate data in the system. Then we can obtain accurate output. Dealing with data before the system begins can unsure system running in read order.

(4) Predicting: according to three stages, we will get the model of predicting average response time of cloud service based on the MGM (1, N) - BP neural network.

3. The Predicted Model

According to running service on the VM, the intensity of the load may change. Service average response time may change. There will be many noise data in the MGM(1,N) predicted model. The grey model cannot predict shifty data in the system. It may appear error to reduce prediction accuracy. To improve prediction accuracy, residual error correction is needed in the system. Although residual identification method may improve prediction accuracy, it cannot apply the grey model. BP neural network is predicted method that is effective and non-linear. It can dispose data that is random, non-linear and wrong data. Its predicted effect is good for changing data. It can improve prediction accuracy and adaptive capacity of predicted model. It can establish function relation by providing data variable. It do not need to know distribution of parameter. Residual error correction of service average

response time can be obtained by BP neural network. BP neural network may improve prediction accuracy. The network structure of BP neural network is as follows in Figure 3.1.

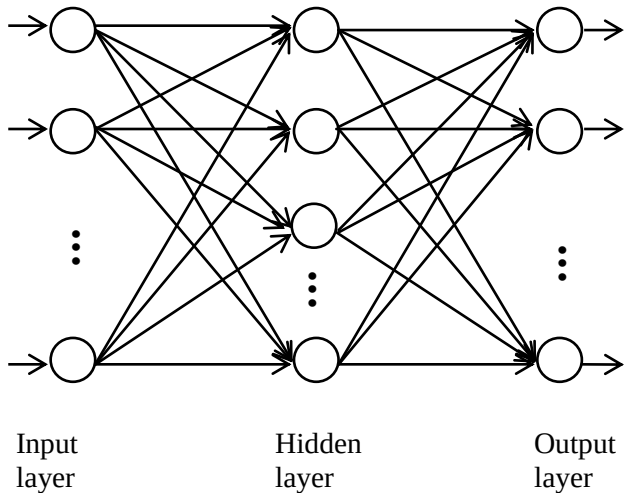


Figure 3.1. The Network Structure of BP Neural Network

The learning process of BP neural network is as follows:

(1) Selecting a group of training sample. Each training sample has two parts: input information and output results.

(2) Selecting a training sample in a group of training sample. Input information is inputted to the input layer.

(3) Calculating output of each layer.

(4) Calculating network error of actual output and desired output.

(5) Calculating data from output layer to the first hidden layer. Adjusting weight of each neuron by reducing error.

(6) Repeating (3)-(5) for each training sample in a group of training sample. It is over until error is acceptable.

The steps of the predicting model are: the average response time and historical data of CPU, memory, network bandwidth are used to construct the forecasting model, namely the MGM(1,N) model; the real value subtracts the value had predicted, so we can get residual error sequence of average response time; we uses BP neural network to forecast the error sequence of the average response time; to get the final forecast, it needs to revise the predicted sequence of original service average response time. The prediction model construction of the average response time process is as follows.

(1) The steps of the structure model

Step1: N – 1 pieces of factor that monitored of average response time and the historical datum of average response time are as the original sequence, remembering:

$$X_i^{(0)} = \{X_i^{(0)}(1), X_i^{(0)}(2), \cdots\cdots, X_i^{(0)}(n)\} \quad i = 1,2,3\cdots n \qquad (3.1)$$

Step2: The original sequence of the CPU, memory, and network bandwidth adds the original sequence of the average response time, getting the trend of average response time and service.

$$X_i^{(1)}(k) = \sum_{j=1}^{k} X_i^{(0)}(j) \quad i = 1,2,\cdots n \tag{3.2}$$

So it shows:

$$X_i^{(1)}(k) - X_i^{(1)}(k-1) = X_i^{(0)}(k) \quad i = 1,2,\cdots n; k = 1,2,\cdots N \tag{3.3}$$

Step3: Setting up N first-order differential equations, getting the changed trend of each index and the average response time:

$$\begin{cases} \dfrac{dX_1^{(1)}}{dt} = a_{11}X_1^{(1)} + a_{12}X_2^{(1)} + \cdots + a_{1n}X_n^{(1)} + b_1 \\ \dfrac{dX_2^{(1)}}{dt} = a_{21}X_1^{(1)} + a_{22}X_2^{(1)} + \cdots + a_{2n}X_n^{(1)} + b_2 \\ \vdots \\ \dfrac{dX_n^{(1)}}{dt} = a_{n1}X_1^{(1)} + a_{n2}X_2^{(1)} + \cdots + a_{nn}X_n^{(1)} + b_n \end{cases} \tag{3.4}$$

The matrix is constituted by differential equations, as follows:

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}, \quad B = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

Among them:

$$X^{(0)}(k) = \left[X_1^{(0)}(k), X_2^{(0)}(k), \cdots, X_n^{(0)}(k)\right]^T$$

$$X^{(1)}(k) = \left[X_1^{(1)}(k), X_2^{(1)}(k), \cdots, X_n^{(1)}(k)\right]^T$$

At this point, the system of differential equations (3.3) can be written as the following form:

$$\frac{dX^{(1)}}{dt} = AX^{(1)} + B \tag{3.5}$$

Among it, $X^{(1)} = (X_1^{(1)}, X_2^{(1)}, \cdots X_n^{(1)})$.

Integral transformation principle is used, the new time response function is as follows:

$$X^{(1)}(t) = e^{At}(X^{(1)}(0) + A^{-1}B) - A^{-1}B \tag{3.6}$$

The differential equations are discretized by trapezoidal integral formula and the rectangular integral formula:

$$X_i^{(0)}(k) = \sum_{j=1}^{n} \frac{a_{ij}}{2} (X_j^{(1)}(k) + X_j^{(1)}(k-1)) + b_i \tag{3.7}$$

Among it, $i = 1,2,\cdots,n; k = 1,2,\cdots N$.

Step4: Calculated on the development trend and identification parameters of each index factor such as CPU, memory, and network bandwidth and the average response time, namely the matrix A, B, the introduction of vector Y_i and matrix L.

$$Y_i = \left[X_i^{(0)}(2), X_i^{(0)}(3), \cdots, X_i^{(0)}(N)\right]^T \tag{3.8}$$

$$L = \begin{bmatrix} \overline{X_1^{(1)}}(2) & \overline{X_2^{(1)}}(2) & \cdots & \overline{X_n^{(1)}}(2) & 1 \\ \overline{X_1^{(1)}}(3) & \overline{X_2^{(1)}}(3) & \cdots & \overline{X_n^{(1)}}(3) & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \overline{X_1^{(1)}}(N) & \overline{X_2^{(1)}}(N) & \cdots & \overline{X_n^{(1)}}(N) & 1 \end{bmatrix}_{(N-1)\times(n+1)} \tag{3.9}$$

Among them, $\overline{X_1^{(1)}}(k) = \frac{1}{2}(\overline{X_1^{(1)}}(k) + \overline{X_1^{(1)}}(k-1)), (k = 1,2,\cdots N)$.

If,

$$a_i = \left[a_{i1}, a_{i2}, \cdots, a_{in}, b_i\right]^T \quad i = 1,2,\cdots,n \tag{3.10}$$

$\hat{a}_i$ is estimated value of a_i is received by the least square method.

$$\hat{a}_i = \begin{bmatrix} \hat{a}_{i1} \\ \hat{a}_{i2} \\ \vdots \\ \hat{a}_{iN} \\ \hat{b}_i \end{bmatrix} = (L^T L)^{-1} L^T Y_i \tag{3.11}$$

The identification parameter matrix is received. A is a matrix. There are n lines and n rows in the matrix A. There are n*n element in the matrix A. Each element is a vector that is named by $\hat{a}_i$. $\hat{a}_i$ is estimated value of a_i is received by the least square method. It has been introduced in (3.10).

B is also a matrix. There are n lines and 1row in the matrix B. There is n element in the matrix B.

$$A=\begin{bmatrix} \hat{a}_{11} & \hat{a}_{12} & \cdots & \hat{a}_{1n} \\ \hat{a}_{21} & \hat{a}_{22} & \cdots & \hat{a}_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ \hat{a}_{n1} & \hat{a}_{n2} & \cdots & \hat{a}_{nn} \end{bmatrix}_{n\times n} \quad B=\begin{bmatrix} \hat{b}_1 \\ \hat{b}_2 \\ \vdots \\ \hat{b}_n \end{bmatrix}_{n\times 1} \tag{3.12}$$

Step5: Getting predicting model:

$$\hat{X}_1^{(1)}(k)=e^{A(k-1)}(X_1^{(1)}(1)+\hat{A}^{-1}\hat{B})-\hat{A}^{-1}\hat{B}(k\geq 1) \tag{3.13}$$

Step6: Getting average response time:

$$\hat{X}_1^{(0)}(k)=\hat{X}_1^{(1)}(k)-\hat{X}_1^{(1)}(k-1)(k\geq 2) \tag{3.14}$$

(2) The steps of the correction residual of BP neural network

Step1: Calculating the residual error sequence of the average response time, formula is as follows:

$$\Delta(k)=X_i^{(0)}(k)-\hat{X}_i^{(0)}(k)(k=1,2,\cdots N) \tag{3.15}$$

Step2: To simplify $\Delta(k)$.

$$\Delta(k)=(\Delta(k)-s)/(t-s) \tag{3.16}$$

Among it, $s=(9\Delta(k)_{\min}-\Delta(k)_{\max})/8$; $t=(9\Delta(k)_{\max}-\Delta(k)_{\min})/8$; $\Delta(k)_{\max}$ is max value of the residual error sequence. $\Delta(k)_{\min}$ is min value of the residual error sequence.

Step3: Determining network structure of BP neural network

The network of function is predicting residual error sequence of average response time. The structure that is single input and single output is used in the network. It inputs neurons and outputs neurons. The number of connotative neurons is ensured as follows:

$$l=\sqrt{(m+n)}+a \tag{3.17}$$

Among it, m is the number of inputted neurons, n is the number of outputted neurons, a is constant between 0 and 10.

Step4: Determining training parameters of BP neural network

There are 7 parameters in Table 3.1.

Table 3.1. The Initial Value of Parameters in BP Neural Network

Parameters	Value of parameters
Passing parameters of hidden layer	Sigmod function
Passing parameters of output layer	Purlin function
Training algorithm	Gradient descent algorithm
Learning efficiency	0.05
Momentum factor	0.95
Training error precision	0.0001
Iterations	5000

Step5: Through a lot of trainings, it chooses efficient network structure and saves training results. At last, the residual error sequence $\hat{\Delta}(k)$ is received.

(3) The residual correction of the average response time

The initial forecast $\hat{X}_1^{(0)}(k)$ of average response time is received by MGM(1,N) grey forecasting model. The residual error sequence $\hat{\Delta}(k)$ is received by BP neural network. Then, the initial forecast $\hat{X}_1^{(0)}(k)$ adds the residual error sequence $\hat{\Delta}(k)$. The final forecast $X_1^{(0)}(k)$ of average response time may be received.

$$X_1^{(0)}(k) = \hat{X}_1^{(0)}(k) + \hat{\Delta}(k)(k > N) \tag{3.18}$$

(4) To get predicted results

To get the final results, it needs to revise the predicting sequence of the original service average response time.

4. The Experimental Results

4.1. Experimental Environment

The experimental is performed on IBM servers. According to the needed number of virtual machines, virtual machines are deployed on the server. The hardware configuration is shown in Table 4.1.

Table 4.1. The Hardware Configuration of IBM Server

Hardware indicators	Configuration
CPU	4 nuclear Intel XeonE5506 1.87GHz
Memory	16GB,1333MHz
Hard disk	500G
OS	Linux

The Xen [7] is used to manage and monitor resource that is used by many users on the server. The memory of Xen is 1500M. It is two VCPUs.

The needed servers, virtual machines and configuration of virtual machines are shown in Table 4.2.

Table 4.2.The Virtual Machine Resource Configuration of Services

Service ID	The number of VM	Configuration		
		VCPU	Memory	Bandwidth
1	3	1	1500	20
2	3	1	2000	10
3	2	2	1000	30

4.2. Experimental Process and Results Analysis

The specific steps are as follows:

(1) Data selection

According to above-mentioned analysis about service performance indicators, historical data should be collected as follows: occupancy rate of CPU, memory, disk, network bandwidth, SWAP, average response time. To obtain correlation of many factors, correlation analysis methods should be used. These factors are shown as follows in Table 4.3.

Table 4.3. The Index Factor Relational Degree of the Service Performance

	CPU	Memory	Disk	Network bandwidth	SWAP
Correlation	0.9173	0.5336	0.4172	0.5599	0.3157

(2) Data processing

(3) Construct MGM(1,N) model

(4) The residual error of neural network correction

(5) Service average response time is obtained by average method on the VM

The experimental analysis MGM(1,N)-BP neural network algorithm, GM(1,1) algorithm and MGM(1,1) algorithm, then compares experimental results. The results are shown in Table 4.4 and Table 4.5.

Table 4.4. Comparison Experiment of Three Kinds of Prediction Algorithm

Time interval	Real value	Predictive value		
		GM(1,1)	MGM(1,N)	MGM(1,N)-BP
41	189.88	177.36	180.67	186.21
42	200.31	180.07	182.55	190.11
43	168.93	182.82	184.39	180.13
44	200.50	185.62	186.17	188.27
45	210.87	188.45	187.91	185.38

Table 4.5. Error Comparison Results of the Three Prediction Algorithm

Algorithm	Average relative error	Mean absolute percentage error	Mean error
GM(1,1)	15.326	12.103%	0.121
MGM(1,N)	12.134	8.988%	0.089
MGM(1,N)-BP	10.527	7.561%	0.072

5. Conclusion and Outlook

The paper puts forward the method of predicting average response time of cloud service based on the MGM (1, N) - BP neural network. The method improves previous research results, combines with two predicting methods and achieves better prediction ability. First, we analyze factor that affect service average response time and disposes relevant indicators. Then, we elaborate the method that based on the MGM (1, N) - BP neural network of average response time. Finally, we verify the feasibility and accuracy through experiments. In the next step of research, MGM (1, N) - BP neural network predicting method should combine with regression predicting methods which can increase the predicted accuracy and shorten the predicted time.

Acknowledgements

This work was supported by the Fundamental Research Funds for the Central Universities(No.120204003) and the NSFC Major Research Program(61100090).

References

[1] J. J. Briedé Deferme, S. M. H. Claessen, D. G. J. Jennen, R. Cavill and J. C. S. Kleinjans, "Time series analysis of oxidative stress response patterns in HepG2: A toxicogenomics approach", Toxicology, vol. 306, **(2013)** April 5, pp. 24-34.

[2] M. Mishra, A. Das and P. Kulkarni, "Dynamic resource management using virtual machine migrations", Communications Magazine, IEEE, vol. 50, no. 9, **(2012)**, pp. 34-40.

[3] J. Gu, J. Hu, T, Zhao and G. Sun, "A new resource scheduling strategy based on genetic algorithm in cloud computing environment", Journal of Computers, vol. 7, no. 1, **(2012)**, pp. 42-52.

[4] L. Shuran and L. Shujin, "Applying BP Neural Network Model to Forecast Peak Velocity of Blasting Ground Vibration", Procedia Engineering, vol. 26, **(2011)**, pp. 257-263.

[5] J. Lin and R.-J. Lian, "Design of a grey-prediction self-organizing fuzzy controller for active suspension systems", Applied Soft Computing, vol. 13, no. 10, **(2013)** October, pp. 4162-4173.

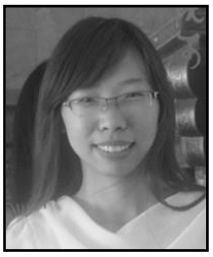
[6] S.-H. Yang and Y.-P. Chen, "An evolutionary constructive and pruning algorithm for artificial neural networks and its prediction applications", Neurocomputing, vol. 86, pp. 140-149, **(2012)** June 1.

[7] A. Menon, J. R. Santos and Y. Tumer, "Diagnosing performance overheads in the xen virtual machine environment", Proceedings of the First ACM/USENIX International Conference on Virtual Execution Environments, **(2005)**.

Authors

Jun Guo, he received his Ph.D. in Northeastern University, China. Jun is a associate professor in Northeastern University. His main study is software testing and optimizing performance on the cloud.

Qun Ma, he is a master in Northeastern University, China. His main study is optimizing performance on the cloud.



Qingmin Ma, she is a master in Northeastern University, China. Her main study is optimizing performance on the cloud.

Yongming Yan, he received his MS degree in software engineering from the Northeast University in 2007. Now he is a PhD candidate at the College of Information Science and Engineering, Northeastern University. His current research interests include cloud computing and service optimization.

Qingliang Han, he is a associate research fellow in Shandong Computing Center. His research interesting is software testing, cloud computing, information security.

