Predicting Virtual Machine Resource Consumption Based on Optimized Grey Model

Shi Aiwu, Gao Di and He Kai
Cloud Computing and Big Data Research Center, Wuhan Textile University, Wuhan 430200, China
451427338@qq.com

Abstract. In order to relatively change the computing specifications of virtual machines (VMs for short) for the changing resource consumption of VMs, and reduce the impact of the changes, this paper proposes the optimized grey GM(1, 1) model to predict the resource consumption of VMs. Because VM resources may fluctuate greatly in a short time, the initial data are smoothed first, and then modeled, the model edge value is optimized according to the characteristics of cloud environment. Experiments show that the accuracy and stability of the prediction model can be improved by sorting out the initial data and optimizing the model. The prediction value is helpful to design the allocation scheme of VM resources, improve the utilization rate of physical resources in cloud environment.

1. Introduction
Besides many large enterprises, many small and medium-sized enterprises begin to deploy their own private cloud. Due to the different loads of VMs in different periods, the consumption of VM resources varies greatly. In order to improve the utilization of physical resources, it is necessary to adopt appropriate schemes to adjust the computing specifications of VMs[1,2]. However, the change of computing specification of VM is a complicated process, the performance of VM is unstable when it changes due to delay[3]. In order to solve these problems in real production environment, this paper uses GM(1, 1) grey prediction model to predict the VM resource consumption at the next moment. GM(1, 1) model is the most widely used model in grey prediction, Xu Huafeng and Fang Zhigeng rebuild the whitening equation by analytic deformation of the model on the mathematical prototype, and proposes a new whitening equation[4]. Cheng Maolin optimizes its solution from the whitening equation, and proposes a three-sum and three-point method from the grey derivative[5]. Shu Fuhua and Song Liangmei optimize the grey derivative and background value to achieve unbiased fitting of non-homogeneous exponential function, and solves the problem of data with non-homogeneous exponential characteristics to a certain extent[6]. Meng Fanlin attempts to use particle swarm optimization algorithm (POS) to calculate model parameters, and uses the similarity between the original sequence of grey correlation analysis measurement model and the fitting sequence[7]. These methods improve the accuracy of the model in the corresponding range, but unsuitable for the prediction of VM resource consumption in cloud environment.

This paper mainly optimizes GM(1,1) model from two aspects. Firstly, according to the characteristics of VM resource changes, the original sequence is smoothed and reconstructed. Secondly, it uses the idea of least squares to modify the boundary value for the solution of whitening equation to ensure that the total error of prediction is minimum. The optimized model is more suitable to predict VM resource consumption in cloud environment.
2. Prediction model

This chapter mainly describes the process of building a virtual machine resource consumption prediction model. Firstly, the traditional grey prediction GM(1, 1) model is introduced. Then an improved GM(1, 1) model is proposed for the characteristics of virtual machine resource consumption in cloud environment.

2.1 Grey GM(1, 1) Model

Grey prediction uses the original data to generate regular data series to establish the relevant differential equation model, and then predicts the development trend of things. The traditional GM(1, 1) modeling process is as follows.

Let’s assume the primitive sequence \( X^{(0)}(k) = (x^{(0)}(1), x^{(0)}(2), \ldots, x^{(0)}(n)) \). Generally, the method of cumulative generation is used to generate grey series \( X^{(1)} = (x^{(1)}(1), x^{(1)}(2), \ldots, x^{(1)}(n)) \).

\[
x^{(1)}(k) = \sum_{i=1}^{k} x^{(0)}(i), \quad k = 1, 2, \ldots, n
\]

If \( z^{(1)}(k) \) is generated by the adjacent value of the grey sequence, there are:

\[
z^{(1)}(k) = \alpha x^{(1)}(k) + (1-\alpha)x^{(1)}(k-1), \quad k = 2, 3, \ldots, n
\]

The value of alpha is generally 0.5.

Then we define the grey differential equation of GM(1, 1) model as follows:

\[
x^{(0)}(k) + ax^{(1)}(k) = b
\]

In the formula, \( x^{(0)}(k) \) is called grey derivative, \( a \) is called development coefficient, \( z^{(1)}(k) \) is called albino background value, and \( b \) is called grey action quantity. Since \( x^{(0)}(k) \) and \( z^{(1)}(k) \) are known, we can solve the \( a \) and \( b \), and bring \( k = 2, 3, \ldots, n \) into formula (3). The values of \( a \) and \( b \) can be obtained by using linear regression of one variable:

\[
H = (a, b)^T = (N^T N)^{-1} N^T D
\]

In formula (4):

\[
D = (x^{(0)}(2), x^{(0)}(3), \ldots, x^{(0)}(n))^T, \quad N = \left[ \begin{array}{cccc}
-z^{(1)}(2) & -z^{(1)}(3) & \ldots & -z^{(1)}(n) \\
1 & 1 & \ldots & 1
\end{array} \right]^T
\]

After calculating the values of \( a \) and \( b \), the whitening differential equation is transformed by the gray differential formula (3). The idea of construction is to treat time \( k = 2, 3, \ldots, n \) as a continuous variable \( t \). The equation is:

\[
\frac{dx^{(1)}(t)}{dt} + ax^{(1)}(t) = b
\]

Solving formula (6) by separating variables:

\[
x^{(1)}(t) = (x^{(0)}(1) - \frac{b}{a})e^{-at} + \frac{b}{a}
\]

Then the prediction value formula of the original sequence is obtained as follows:

\[
x^{(0)}(k+1) = x^{(1)}(k+1)-x^{(1)}(k) = (x^{(0)}(1) - \frac{b}{a})(1-e^{-at}) + \frac{b}{a}, \quad k = 1, 2, \ldots, n - 1
\]

2.2 Optimized GM(1, 1) Model

In the background of resource consumption prediction of VM, an optimized of GM(1,1) model based on its application scenario characteristics is proposed. The definition of the relevant parameters used is shown in the table below.

| Parameter | Definition |
|-----------|------------|
| R(k)      | Real resource consumption value of virtual machine at time k |
| R^(0)(k)  | The original value of the prediction model of virtual machine at time k |
The grey value of the forecasting model of virtual machine at time k
The grey neighbor value of prediction model of virtual machine at Time k
Development Coefficient of Grey Model
Grey Action Quantity of Grey Model
Edge Correction Value of Grey Model

GM(1, 1) model has the characteristics of fast operation speed and small requirement for original data. However, it is not applicable to all the original data, generally applicable to the original sequence which does not fluctuate sharply or contains the law of exponential change. For the resource consumption of virtual machines, especially for CPU, the fluctuation of operating system in stable state is small, but when the load changes suddenly, it causes a sudden increase (or decrease). In order to minimize the impact of load mutation on prediction accuracy. The original data is smoothed and then input as the initial value of the model.

We use the method of weights balance to smooth four consecutive moments. Considering that the closer the time is, the greater the influence should be, the weights should be allocated according to the distance of the time. The specific formula is as follows.

\[ R^{(0)}(k) = 0.1R(k-3) + 0.2R(k-2) + 0.3R(k-1) + 0.4R(k), \quad k = 1,2,...,n \]  

Thus, we construct a new original sequence \( X(0) = (R^{(0)}(1), R^{(0)}(2), \ldots, R^{(0)}(n)) \) from the original data, and then generate grey sequence \( X(1) = (R^{(1)}(1), R^{(1)}(2), \ldots, R^{(1)}(n)) \) from the newly formed original sequence by the way of cumulative generation. In this formula:

\[ R^{(1)}(k) = \sum_{i=0}^{k} R^{(0)}(i), \quad k = 1,2,...,n \]  

The constructable differential equation is:

\[ R^{(0)}(k) + aZ^{(1)}(k) = b, \quad k = 2,3,...,n \]  

In this formula:

\[ Z^{(1)}(k) = 0.5R^{(1)}(k) + 0.5R^{(1)}(k-1), \quad k = 2,3,...,n \]  

The values of \( a \) and \( b \) can be solved by the formula (5). The prediction value of \( R^{(1)}(k) \) can be obtained by solving the albino differential formula (6):

\[ \hat{R}^{(1)}(k+1) = (R^{(1)}(1) - \frac{b}{a})e^{-ak} + \frac{b}{a}, \quad k = 1,2,...,n-1 \]  

Therefore, the prediction value of \( R^{(0)}(k) \) can be obtained as follows:

\[ \hat{R}^{(0)}(k+1) = \hat{R}^{(1)}(k+1) - \hat{R}^{(1)}(k) = (R^{(1)}(1) - \frac{b}{a})(1 - e^{-ak}) + \frac{b}{a}, \quad k = 1,2,...,n-1 \]  

It can be seen from Formula (13) that the boundary value \( R^{(1)}(1) \) has a great influence on the prediction results of the model. In the traditional GM(1, 1) model, as shown in Formula (8), it is assumed that \( R^{(1)}(1) = R^{(0)}(1) \). Here we modify the boundary value \( R^{(1)}(1) \) by optimizing \( R^{(1)}(1) = R^{(0)}(1) + u \). With the purpose of ensuring the minimum prediction error, the idea of least squares is adopted, that is to say, the following formula is valid:

\[ \min_u \sum_{k=2}^{n} (R(k) - \hat{R}(k))^2 \]  

Solving formula (8) can be obtained:

\[ u = \frac{\sum_{k=1}^{n} R^{(0)}(k)e^{-ak(k-1)} - R^{(0)}(1) + \frac{b}{a}}{(1-e^{-2an})(1-e^{-2a})} \]  

The prediction formula can be obtained by introducing formula (15) into formula (13):

\[ \hat{R}^{(0)}(k+1) = e^{-ak}(1 - e^{-2an}) \frac{\sum_{k=1}^{n} R^{(0)}(k)e^{-ak(k-1)}}{1-e^{-2an}} + \frac{b}{a}, \quad k = 1,2,...,n-1 \]
The final resource consumption prediction formula for the next moment of VM can be obtained by introducing formula (16) into formula (9):

$$
\hat{R}(k+1) = \frac{e^{-ak}(1-e^{-2ak})}{1-e^{-ak}} \sum_{k=1}^{\infty} R^{(0)}(k) e^{-ak(k-1)} + \frac{b}{a} \frac{R(k-2) + 2R(k-1) + 3R(k)}{10}, \; k = 1, 2, \ldots, n-1
$$

(17)

3. Simulation Experiment

This paper uses open source IaaS ZStack to build a private cloud platform, and uses KVM technology to create three virtual machines above to verify the effects of the model.

3.1 Experimental Environment

In the ZStack cloud platform, three virtual machines are created on a physical server. Because the fluctuation of CPU resources is the most intense, the CPU resources are predicted as an example. The VCPU is used as a unit to predict resources, and the ratio of VCPU to physical CPU is 10:1. Only VCPU is different in the calculation specifications of the three virtual machines, and other parameters are guaranteed to be consistent, as shown in Table 2.

|               | VM1               | VM2               | VM3               |
|---------------|-------------------|-------------------|-------------------|
| System Image  | Win 10 64-bit     | Win 10 64-bit     | Win 10 64-bit     |
| VCPU          | 40                | 60                | 80                |
| Memory        | 4 G               | 4 G               | 4 G               |
| Storage       | 60 G              | 60 G              | 60 G              |
| VMM           | KVM               | KVM               | KVM               |

3.2 Experimental Design

Experiments are carried out from three aspects: the initial observation data, the load of each virtual machine, and the selection of the prediction model. Specifically formulate the following three rules.

1. We choose the same time period to observe the VCPU consumption of three virtual machines. The specific scheme is to observe the VCPU consumption in the same minute with an observation interval of 1s. A total of 60 observations per virtual machine are used for model calculation, and 61 seconds of observations are recorded for comparison of prediction results.

2. Considering three common situations encountered in actual prediction, we compare the prediction effect of the model in various situations. The specific scheme is to apply nearly the same load to the three virtual machines before the observation data, and then to start the observation after their stabilization. In VM1, we do not operate at any time during the observation period, In VM2, we run an application at some time during the observation period, and then it does not operate any more after the operation of the application. In VM3, we run an application at some time during the observation period and run another application at the end of the observation period.

3. In order to further verify the pros and cons of the prediction model, we select three kinds of prediction models for comparative analysis.

With the rule 1 and 2, The VCPU consumption of the three virtual machines we observed is shown in figure 1.
Figure 1. VCPU Consumption Observation of Virtual Machine.

Figure 2. Relative Residual of Traditional GM(1, 1) Model.

Figure 1 shows the VCPU consumption characteristics of the three virtual machines. VM1 remained stable throughout the process. VM2 fluctuated during the period and finally remained stable. VM3 began to remain stable, fluctuated in the middle, and rose sharply at the last minute. Next, according to rule 3, three prediction models are selected to predict the observed values.

3.3 Traditional GM(1, 1) Model

Instead of smoothing the observed values in figure 1, we can directly predict them as the original sequence $X^{(0)}$. Combined with Formula (5) and Formula (8), we can solve the VCPU prediction values of each virtual machine. For the sake of reflecting the prediction accuracy more intuitively, the relative residual of the predicted values $\varepsilon$ is introduced here, which is defined as follows:

$$\varepsilon(k) = \frac{R(k) - \hat{R}(k)}{R(k)}$$

(18)

If the absolute value of relative residual $|\varepsilon| < 0.1$, we can see the prediction result more accurately.

In principle, we only need to focus on the relative residuals at the end of the observation time, $k = 61$. From Formula (5), we can see that the values of model parameters $a$ and $b$ are related to all the values in the original sequence. It is improper for us to use the prediction model derived from the original sequence of length 60 to predict the VCPU consumption of the virtual machine at the time of $k=i(i<=60)$. However, the inappropriate predictions can still reflect the characteristics of the model to a certain extent. For example, the stability of the model can be obtained by observing the trend of relative residuals. So we still use the obtained model to calculate the relative residuals at each time. The final results are shown in figure 2.

As can be seen from figure 2, for the traditional GM(1, 1) model at $k=61$, both VM1 and VM2 can achieve high accuracy, while VM3 barely meets the requirements which has a larger prediction error. From the overall trend of relative errors, VM1 has always maintained a high prediction accuracy. VM2 has a large prediction error at the beginning, but with the increase of time, the prediction accuracy gradually improves and finally tends to be stable. The prediction accuracy of VM3 is extremely unstable, especially when the medium-term load suddenly increases, it is especially prominent, and only gradually decreases at the last moment.

3.4 GM(1, 1) Model after Smoothing Data

Next, we use GM(1, 1) model to predict the observed data directly after smoothing. Note that after smoothing, the original sequence length changes from 60 to 57, and the prediction time starts from $k=5$. The combined formula (10), formula (12) and formula (14) (where $R^{(i)}(0) = R^{(0)}(0)$) can be used to obtain the VCPU consumption prediction values of each virtual machine at this time. The relative residual is obtained by formula (19) as shown in the figure 3.
From figure 3, after smoothing data, it can be seen that the overall trend of relative residuals of the three virtual machines remains unchanged, but the stability has been greatly improved. The relative residuals of VM1 remained below the pass line, but their mean and variance decreased significantly. The relative residuals of VM2 tend to be stable more quickly, and the mean and variance of VM2 decrease significantly in the second half of the time. Although VM3 fluctuates obviously, its regularity is stronger and shows a significant decreasing trend with the increase of time. This shows that smoothing data is obviously helpful to improve the stability and prediction accuracy of GM(1, 1) model.

3.5 Optimized GM(1, 1) Model

Finally, we smoothed the observed data and used the optimized GM(1, 1) model to predict. The combined formula (12) and formula (18) can be used to obtain the VCPU consumption prediction values of each virtual machine at this time. The relative residual is obtained by formula (19) as shown in the figure 4.

Figure 4 is almost the same as figure 3. It shows that the optimized GM(1, 1) model does not change the change trend of the original GM(1, 1) model, but the relative residual of the optimized GM(1, 1) model at most times is better than the original model, and the mean and variance of the relative residual are reduced.

3.6 Comparison and Conclusion of Prediction Accuracy

The model mentioned above can only be applied to the prediction at $k=61$, but with the purpose of observing the stability of the model, we calculate the predictions at other times according to the model. The closer the $K$ value is to 61, the smaller the impact of the model on the prediction value. With the purpose of showing the prediction accuracy of the three models more intuitively, we choose the absolute value of the relative residual error at three times $k=59$, 60 and 61 as the criterion of the prediction accuracy of the model. That is, the smaller the accuracy of the variable model is, the higher the accuracy of the prediction model is. Definitions are as follows:

$$\lambda=0.2\varepsilon(59) + 0.3\varepsilon(60) + 0.5\varepsilon(61)$$

(19)

The values of the three VMs under the three models are calculated as the figure 5.
Figure 5. Comparison of Prediction Accuracy

From figure 5, the prediction accuracy of the three models in VM1 is very close and keeps a high accuracy all the time. The three models in VM2 differ greatly from VM1, but the three models are still close. After smoothing the data in VM3, the accuracy of the model is greatly improved.

Based on the previous experimental results, the following conclusions can be drawn: For virtual machine resource prediction in cloud environment, data smoothing can improve the stability and accuracy of prediction, but it has limited improvement for the situation where the load has become stable at the prediction time, and the effect is very obvious for the situation where the load is unstable at the prediction time. In three load environments, the optimized prediction model has slightly improved the stability and prediction accuracy of the original model.

4. Conclusion

This paper proposes an optimized GM (1, 1) model to predict VM resource consumption, there are two main improvements. Firstly, according to the characteristics of virtual machine resource consumption in cloud environment, the observed data are smoothed and modeled. Secondly, an optimized GM (1, 1) model is proposed by modifying some parameters of the model. Our experiment result verify the optimization effect of our proposal. However, the stability of the prediction model is not ideal and the accuracy of the model needs to be further improved. Therefore, in the future work, we will further optimize the model by other ideas.

Reference:
[1] Shen Yao, Qin Xiaolin and Bao Zhifeng 2017 China. J. Soft. 28 579–97.
[2] Kefayat M, Ara AL and Niaki SAN 2015 Energy Convers. Manage. 92 149–61.
[3] Meena J, Kumar M and Vardhan M 2016 IEEE Access. 4 5065–82.
[4] Xu Huafeng and Fang Zhigeng 2011 China. Maths. Practice Theory. 41 163–67.
[5] Cheng Maolin 2018 China. J. Suzhou. Univ. Sci. Technol Nat. Sci. 35 6–10.
[6] Shu Fuhua and Song Liangmei 2018 China. J. Shanghai. Business. Univ. 19 15–21.
[7] Meng Fanlin, Wang Tianhui and Li Bingjun 2017 IEEE International Conference on Grey Systems and Intelligent Services. pp 154-158.