International Journal of Computer Applications Technology and Research
Volume 8–Issue 01, 29-32, 2018, ISSN:-2319–8656

Research on Cloud Hard Disk Capacity Prediction Scheme Based on Time Series Model

Zizhen Yuan
School of Communication Engineering
Chengdu University of Information Technology
Chengdu, China

Chengyu Wen
School of Communication Engineering
Chengdu University of Information Technology
Chengdu, China

Xiaoli Zhang
School of Communication Engineering
Chengdu University of Information Technology
Chengdu, China

Abstract: OpenStack is a cloud operating system, but it does not provide the predictive functionality for cloud disk usage of the virtual machines running on it. In this paper, it propose a prediction schem of virtual machine cloud hard disk capacity. First the Libvirt API is used to obtain the disk data of the virtual machine. After data processing, the ARIMA time series model is built. Then the capacity of the disk in the future is predicted. This scheme can predict the disk capacity of virtual machine, facilitate the user to expand capacity or issue an alarm, and then reduce disk failure.

Keywords: capacity prediction; time series model; cloud hard disk; OpenStack; Elastic block storage

1. INTRODUCTION

OpenStack is an open source private cloud platform. Due to it's excellent features, rich functions and free features as open source software, a large number of researchers and companies are contributing code to it, and it has a secondary development as their private cloud platform by various manufacturers around the world. OpenStack Cinder is a block storage component in OpenStack. Its main function is to conduct logical abstraction between virtual machine and heterogeneous storage as a unified driver management, ensure the high availability of storage services, and provide block storage scheduling services for virtual machine through it's own scheduling algorithm. Specific ways of storage are multiple including local storage, Ceph, GlusterFS, etc. Current block storage does not provide capacity prediction function for users. Users need to manually predict the future capacity of disk and perform capacity expansion operation, which may cause system failure due to insufficient disk space caused by overdue capacity expansion. This paper proposes a scheme that can monitor the disk usage data of virtual machine and model the data through time series model, so as to predict the space that the storage may grow in the future. Later users can choose to issue disk warning or perform capacity expansion operation.

2. DATA MONITORING MODULE

The virtual machine monitor solutions can be realized by a built-in monitoring module in a virtual machine, or it's own monitoring component Ceilometer, or the Libvirt library[1]. The built-in monitoring module in the virtual machine can achieve better monitoring effect, but the service needs to be turned on in the host. If the user manually disables the service, the data cannot be collected, and there are too many uncontrollable factors, so this method is abandoned. OpenDtack's own monitoring component Ceilometer can obtain the data about CPU of virtual machine, disk read-write and network read-write. But there is no disk space data, which is one disadvantage. The other point is that the request timeout is too serious to collect data in real time. Therefore, OpenStack's built-in virtualization software Libvirt was selected to obtain the virtual machine disk data.

Libvirt provides a native layer interface for KVM in OpenStack cloud platform, which can perform basic management operations on virtual machines in the cloud platform. The Libvirt library is implemented in C and includes direct support for python, which is the python language binding toolkit based on the Libvirt API[2]. This article can achieve the collection of virtual machine performance indicators through the functions in the libvirt-python package.

Because the libvirt-python package does not directly capture the function of CPU utilization of the virtual machine, the acquisition of cpu_usage in the virtual machine performance index requires other methods. In this paper, the cpu_numbers and cpu_time values were obtained by using domain.info() function, and cpu_usage was calculated by combining with time.time() function[3]. The function of domain.info() and time.time() needs to be run simultaneously before the time.sleep() runs or after it does. Domain.info() is run to get cpu_numbers, cpu_time_1 and cpu_time_2, and time.time() is run to get t_1 and t_2. *cpu_usage* was obtained through calculation of the obtained value. The calculation formula is shown in following :

$$\text{cpu\_usage} = 100 \times (t_1 - t_2)/((\text{cpu}_{\text{time1}} - \text{cpu}_{\text{time2}}) \times \text{cpu\_numbers} \times 10^9)$$

Memory performance indicator data in virtual machine performance indicators is mainly collected through domain.memorystats () function. However, mem_usage cannot be directly obtained from a function, and it can only be calculated by using the available and unused data collected by domain.memorystats () function. Major_fault can be directly collected by using domain.memorystats()[4]. The specific formula is as follows:

$$\text{mem\_usage} = (\text{available} - \text{unused})/\text{available}$$

Disk I/O performance index of the virtual machine can also be collected through domain.blockstats().

Network performance indexes of virtual machines can be collected through domain.interfacestats().

3. MODEL BUILDING

3.1 Date Processing

In real business, the monitoring system collects disk information at regular intervals every day, but in general, the disk capacity attribute is a fixed value (the case of midway expansion is not considered), so there will be duplicate data of disk capacity in the original data of the disk. In the process of data cleaning, duplicate data of disk capacity are removed, and the disk capacity of all servers is taken as a fixed value to facilitate model warning.

In disk data after data cleaning, disk-related properties exist in the data as records in kilobytes. Because the disk information of each server can be distinguished by the NAME, ID and ENTITY attributes in the table, and the above three attribute values of each server are unchanged, the values of the three attributes can be combined to construct new attributes[5].

3.2 Model Building

An application failure is usually not a sudden crash (unless the server is directly cut off), but rather a gradual process.For example, if the system runs for a long time, the data will continue to be written to the storage, and the storage space will gradually decrease, and eventually the disk will be full and the system will fail.Therefore, when human factors are not considered, the storage space has a strong correlation with the change of time, and historical data has a certain impact on the future development, so we can use the time series analysis method to build the model[6].

First, the stability test of the observation value sequence is required. If the observation value sequence is not stable, the difference processing is carried out until the data after the difference processing is stable. After the data is stable, the white noise is tested. If the model does not pass the white noise test, the model identification is carried out to identify which model it belongs to, AR, MA or ARMA[7]. Furthermore, determine the order of the model by BIC information criterion, then determine p and q parameters of ARIMA model. After model recognition, model test is required to detect whether the model residual sequence is the white noise sequence. If the model does not pass the test, it needs to be re-identified. The maximum likelihood estimation method is used to estimate the parameters of the tested model. Finally, use the model to predict and compare the actual value and the predicted value for error analysis[8]. If the error is relatively small (the error threshold needs to be set through business analysis), it indicates that the model fitting effect is good, and the model can be ended. Instead, the parameters need to be reevaluated.

The following methods are required in the process of model building.

1) stationarity test: in order to determine that there is no random trend or definite trend in the original data sequence, stationarity test should be carried out on the data, otherwise the phenomenon of "pseudo-regression" will occur.This paper adopts the method of unit root test (ADF) or the method of sequence diagram to test the stationarity.

2) white noise test: in order to verify whether the useful information in the sequence has been extracted, it is necessary to carry out white noise test on the sequence.If the sequence test is a white noise sequence, it means that the useful information in the sequence has been extracted, and all the rest are random perturbations that cannot be predicted and used[9]. There are currently two commonly used methods, Q statistics:

$$Q = n\sum_{k=1}^{m} \widehat{\rho_k^2}$$

and LB statistics after correction:

$$LB = n(n+2)\sum_{k=1}^{m}\left(\frac{\widehat{\rho_k^2}}{n-k}\right)$$

3) model identification: maximum likelihood ratio method is adopted to estimate the model and estimate the values of various parameters.Then, for each different model, BIC information criterion is used to determine the order of the model, and p and q parameters are determined, so as to select the optimal model.

4) model test: after the model is determined, check whether the residual sequence is white noise.If it is not white noise, it indicates that there is still useful information in the residual, which needs to be modified or further extracted.

5) model prediction: the verified model is applied for prediction to obtain the predicted value in the next 5 days and compare it with the actual value.

6) model evaluation: in order to evaluate the effect of time series prediction model, mean absolute error, root mean square error and mean absolute percentage error can be used[10].

4. EXPERIMENTAL ENVIRONMENT AND RESULTS

4.1 Experimental Environment

In order to verify the prediction effect of the time series model, the following experimental clusters are built, which are respectively composed of a control node, a computing node and a storage node.

Table 1. Experimental environment

Name	Operating system	Hard drive sizes	Memory Sizes
Controller	Centos7	20GB	4GB
Compute	Centos7	100GB	16GB
Block Storage	Centos7	1TB	4GB

4.2 Experimental Results

Data collection is conducted according to the granularity of 5 minutes, and the collected disk data format is shown in table 2:

Table 2. Disk capacity

Name	ID	Entity	Time	Value
CWXT	184	C:\	2018/12/27	34270787.33
CWXT	184	D:\	2018/12/27	80262592.65

Disk data after attribute construction is shown in table 3.

Table 3. Disk data after Attribute construction

Name	184: C:\	184: D:\	Time
CWXT	34270787.33	80262592.65	2018/12/27

Then the stability test is carried out on the data. The test results are shown in table 4:

Table 4. Stationarity test

Name of the sequence	Stationarity	The corresponding p value	Stability after N - order difference
D disk size for use	Non-stability	0.8921	1

After the stationarity test, white noise test is needed to monitor whether there is any useful information.The results are shown in table 5:

Table 5. White noise test

Name of the sequence	White noise	The corresponding p value
D disk	False	$9.9585*10^{-6}$
D disk first order difference	True	0.1143

Then, model recognition of the data sequence is required. The recognition results are shown in table 6:

Table 6. Model recognition

Name of the sequence	Model type	Minimum BIC value
D disk size for use	ARIMA(0,1,1)	1300.46

According to the identified ARIMA(0,1,1) model, its residual sequence has been proved white noise, which has passed the test successfully.

Then the modetested is used to make predictions and the prediction values for the next 5 days are obtained , as shown in table 7:

Table 7. Prediction results

Days	Prediction value	Actual value
1	83.79671	83.20745
2	83.99399	82.95645
3	84.16823	82.66281
4	84.34248	85.6081
5	84.51672	85.23705

The model evaluation results are shown in table 8:

Table 8. Model evaluation

Mean absolute deviation	Mean squared error	Mean absolute percent error
1.0236	1.1621	1.2207

5. CONCLUSIONS

This paper proposes a block storage capacity prediction scheme based on time series, the first data monitoring module was designed,and the Libvirt API can be used to obtain disk usage data of the virtual machine, then to obtain clean data through data cleaning and property construction. Then use these data to predict the disk capacity by the ARIMA time series model , and the prediction values for the next period of time are obtained, but its predictions for long periods of time are not accurate because of the ARIMA sequence dynamic prediction--- in addition that the first valuepredicted is based on a true value , the rest are all further predicted based on the first prediction value.Therefore, it is better for a short-term capacity prediction.

6. REFERENCES

[1] Stokely, Murray, et al. "Projecting disk usage based on historical trends in a cloud environment." Workshop on Scientific Cloud Computing Date 2012.

[2] Ji, Xue, et al. "PRACTISE: Robust prediction of data center time series." International Conference on Network & Service Management 2015.

[3] Box, George E. P, and G. M. Jenkins. "Time series analysis forecasting and control - Rev. ed. " Journal of Time 31.4(1976):238-242.

[4] Hassoun, M. H. "Fundamentals of Artificial Neural Networks." Proceedings of the IEEE 84.6(2002):906.

[5] Lee, Ho Seong, and L. Guo. "Servo performance prediction for high capacity disk drives." American Control Conference 1998.

[6] Gmach, Daniel, et al. "Capacity Management and Demand Prediction for Next Generation Data Centers." IEEE International Conference on Web Services 2009.

[7] Shen, Zhiming , et al. "CloudScale:elastic resource scaling for multi-tenant cloud systems." Acm Symposium on Cloud Computing ACM, 2011.

[8] Gong, Zhenhuan Gong Zhenhuan , X. G. X. Gu , and J. Wilkes . "PRESS: PRedictive Elastic ReSource Scaling for cloud systems." International Conference on Network & Service Management IEEE, 2010.

[9] Meng, Xiaoqiao , et al. "Efficient resource provisioning in compute clouds via VM multiplexing." International Conference on Autonomic Computing DBLP, 2010.

[10] Chen, Yuan Chen Yuan , et al. "Integrated management of application performance, power and cooling in data centers." IEEE/IFIP Network Operations & Management Symposium IEEE, 2010.