Research Article

A Virtual Machine Migration Strategy Based on Time Series Workload Prediction Using Cloud Model

Yanbing Liu,¹ Bo Gong,¹ Congcong Xing,² and Yi Jian¹

¹ Engineering Laboratory of Network and Information Security, Chongqing University of Posts and Telecommunications, Chongqing 400065, China

² Department of Mathematics and Computer Science, Nicholls State University, Thibodaux, LA 70310, USA

Correspondence should be addressed to Yanbing Liu; liuyb@cqupt.edu.cn

Received 6 June 2014; Accepted 19 August 2014; Published 25 September 2014

Academic Editor: Chuandong Li

Copyright © 2014 Yanbing Liu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Aimed at resolving the issues of the imbalance of resources and workloads at data centers and the overhead together with the high cost of virtual machine (VM) migrations, this paper proposes a new VM migration strategy which is based on the cloud model time series workload prediction algorithm. By setting the upper and lower workload bounds for host machines, forecasting the tendency of their subsequent workloads by creating a workload time series using the cloud model, and stipulating a general VM migration criterion workload-aware migration (WAM), the proposed strategy selects a source host machine, a destination host machine, and a VM on the source host machine carrying out the task of the VM migration. Experimental results and analyses show, through comparison with other peer research works, that the proposed method can effectively avoid VM migrations caused by momentary peak workload values, significantly lower the number of VM migrations, and dynamically reach and maintain a resource and workload balance for virtual machines promoting an improved utilization of resources in the entire data center.

1. Introduction

With the rapid and continuous growth of cloud computing on a global scale, typical cloud computing techniques such as virtualization, parallel computation, and distributed database and storage have gained substantial development and been applied extensively in different areas. In particular, as one of the foundational components of cloud computing architecture, virtualization technique plays a critical role in delivering guaranteed cloud computing services [1]. By creating multiple simulating virtual machines (VMs) on the cluster of high performance network servers and providing on-demand services to users via these virtual machines, virtualization is a fundamental technique that can be used to realize the rapid deployment, dynamic allocation, and cross-domain management of IT resources [2–4]. Note however that driven by the constantly changing users' demands, both the number and the workloads of virtual machines vary frequently, which, incidentally, presents a new challenge for resources scheduling and migrations of virtual machines. It has been recognized, by virtue of virtual machine migration

process, that the selections of the source host machine and destination host machine are the most significant steps for virtual machine migrations.

Toward making decisions as to the selections of source and destination host machines, and also avoiding unnecessary virtual machine migrations caused by momentary peak workload values, the so-called live migration strategy for virtual machines is proposed by some researchers [5–8]. Currently, there are two types of virtual machine migration approaches existing in the literature: one is to combine the upper threshold and lower threshold of the host machine to manage the use of resources [9, 10]; the other is to use the workload threshold of the host machine to predict the trend of its subsequent workloads [11–14]. While the former approach is able to resolve the issue of resource waste inflicted by the static workload balancing strategy, it cannot resolve the issue of aggregation conflict which exists in traditional workload balancing strategies. On the other hand, the latter approach is able to resolve the issue of “false alarm” virtual machine migrations caused by some transient peak workload values but fails to put into consideration the uncertainty and

the stochastic nature of the workload values, as well as the combination of both, on host machines.

As such, toward bringing the uncertainty and randomness issue of workload values into the decision process of virtual machine migrations, and thus resulting in a more robust migration strategy, we in this paper propose a new virtual machine migration strategy which is based on the time series predication in cloud theory. This strategy basically works as follows: it first sets up the upper and lower workload thresholds for host machines, then forecasts the future workload tendency of the host machine using cloud theory, and finally stipulates a migration selection criterion and uses this criterion to select the source host machine, destination host machine, and the virtual machine to perform the desired migration. We argue that our proposed virtual machine migration strategy offers a comprehensive treatment for the uncertainty, fuzziness, and randomness of the workload values, converts qualitative notions to quantitative ones and vice versa, eliminates the aggregation conflict problem induced by virtual machine migrations due to some transient and momentary peak workload values, and contributes to obtaining a dynamic balancing of virtual machine resources.

The rest of the paper is structured as follows: Section 2 overviews related work in the literature as to the virtual machine migrations. Section 3 reviews the background knowledge of cloud model and introduces the computation of time series workload prediction. Our proposed virtual machine migration algorithm is presented in Section 4. Section 5 demonstrates the experimental results and analyses of the proposed migration strategy in comparison with other peer works, and Section 6 concludes the paper.

2. Related Work

The subject of virtual machine migration has been extensively studied [15–19]. The primary reason for this is that there is a constant increase in the number of virtual machines at cloud computing data centers, which presents new challenges in terms of the virtual machine resource scheduling and deployment. Due to the fact that the workload of a host machine for virtual machines changes dynamically in accordance with the ever-changing users' service demands, simple and static virtual machine migration strategies are no longer adequate in delivering quality services for users.

Conventional virtual machine migration strategies can be classified into single-threshold method and dual-threshold method [20, 21]. While the single-threshold method only places an upper bound on the workloads of host machines and initiates the virtual machine migration if the workload value is over this upper bound, dual-threshold method places both an upper bound and a lower bound on the workloads of host machines and initiates the migration when the workload is over the upper bound or below the lower bound. Beloglazov et al. [22–25] suggested an adaptive energy-efficient and threshold-based heuristic algorithm which controls the virtual machine migration by monitoring the resource utilization rate. Unfortunately, threshold-based migration strategies lack the ability to foresee the possible

future workload trend of host machines, and consequently may trigger unnecessary and wasteful virtual machine migrations if the workload of the host machine peaks just for a moment (for arbitrary reasons).

Various workload prediction techniques are also used in the context of virtual machine migrations [26–29]. In [28], Khan et al. proposed a hidden Markov model based prediction method with the restriction that the applicability of this method depends on the correlation of time and domain of the workload. Gmach et al. [26] presented a resource pool management strategy on the basis of workload analysis and demand predication but did not address the issue of the actual virtual machine migrations. Zhao and Shen [30] used the autoregressive (AR) model in time series prediction techniques, which predicts future values on the basis of a sequence of n past values ordered in time, to forecast the future workload values. Generally speaking, much of the current research work on prediction techniques fails to relate the workload predication analysis of host machines with the resource management of virtual machines to obtain a more desirable migration strategy.

The main purpose of our work is to improve the existing virtual machine migration strategies by applying the cloud model time series workload prediction technique to the decision procedure and process in virtual machine migrations.

3. Time Series Workload Predication Based on Cloud Model

Cloud model was proposed by Li et al. [14] in 2000, which deals with the conversion between qualitative concepts and quantitative descriptions subject to the notion of uncertainty. There exists a certain kind of mapping between the generally ambiguous describing ability of any natural languages and what objectively exists in the world and is intended to be described by the natural languages. It is interesting to note that this mapping is matched, in a primitive manner, by the essence of the cloud model.

3.1. Cloud Model Basics. Let U be a quantitative domain of precise values and C be a qualitative concept over U . For any $x \in U$, there exists a random number $\mu(x)$ with a stable tendency, which represents the relevance of x with respect to the concept C . The distribution of x over the domain U is called a cloud. Each x corresponds to a cloud droplet $\text{drop}(x, y_i)$. A cloud can be quantitatively characterized by 3 numerical values: expectation E_x , entropy E_n , and hyper entropy H_e (see Figure 1), where

- (i) E_x denotes the most typical quantitative expectation for the qualitative concept,
- (ii) E_n indicates the uncertainty of the concept. The value of E_n shows the range of U over which the concept can be accepted (with distinct uncertainty),
- (iii) H_e is the uncertainty measure of E_n and is affected by both the randomness and the fuzziness of E_n . The value of E_n indirectly indicates the cohesion degree of cloud droplet $\text{drop}(x, y_i)$.

FIGURE 1: The numerical characteristics of cloud model.

A massive amount of cloud droplets gives rise to a cloud. Each droplet represents the phenomenon that the qualitative concept can be quantified to a point in U . According to the cloud generating mechanism in cloud model, there are forward cloud generators and backward cloud generators [31].

As shown in Figure 2(a), cloud droplets (a quantitative notion) are produced by the forward cloud generator using the 3 numerical values E_x , E_n , and H_e which are the characterization of a cloud (a qualitative notion). Every produced droplet is a concrete realization of the qualitative cloud. Conversely, as shown in Figure 2(b), backward cloud generator reverses the process of forward cloud generator by converting cloud droplets $\text{drop}(x_i, y_i)$ into the 3 numerical values E_x , E_n , and H_e (which are an abstract representation of a qualitative cloud).

3.2. The Computation for Time Series Workload Predication. The procedure to obtain a time series workload prediction is depicted in Figure 3. There are four core modules in this procedure: parameter setting/optimization, data preprocessing, prediction modeling by cloud model, and model evaluation, with data preprocessing and prediction modeling by cloud model being the primary ones. The setting of parameters is extremely important as it may affect the accuracy of the prediction results. Collected sample data will be first processed into standardized form via zero-mean operation, difference operation, and center compression operation and then fed into the prediction module to get the preliminary result. Final prediction result will be obtained by performing reversed difference operations on the preliminary prediction result.

Specific steps for computing the workload prediction are given as follows.

Step 1. Let $\{y_t\}$ ($t = 1, 2, \dots, n$) be a known workload sequence indexed by time. Process $\{y_t\}$ by performing zero-mean operation on $\{y_t\}$ to obtain the sequence $\{y'_t\}$. Then

calculate the first-order difference $\Delta y'_t$ of y'_t and the second-order difference $\Delta^2 y'_t$ of y'_t as follows:

$$\begin{aligned}\Delta y'_t &= y'_{t+1} - y'_t, \\ \Delta^2 y'_t &= \Delta y'_{t+1} - \Delta y'_t.\end{aligned}\quad (1)$$

We use $\{D_t\}$ ($t = 1, 2, \dots, n-2$) to denote $\{\Delta^2 y'_t\}$ in subsequent steps.

Step 2. Suppose we need to predict k subsequent workload values. Using the deviation technique in nonlinear data standardizations, standardize $\{D_t\}$ obtained in Step 1 as follows:

$$F_t = \frac{D_t - \overline{D_t}}{\delta}, \quad (2)$$

where $\overline{D_t}$ is the average of $\{D_t\}$ and δ is the standard deviation of $\{D_t\}$.

Step 3. Let $F_{\max}$ be the maximum value of $\{F_t\}$ calculated in Step 2, and μ_t be the ratio of F_t with $F_{\max}$. Process $\{F_t\}$ by eliminating all F_t 's in $\{F_t\}$ whose $\mu_t > 0.9999$; call the result $\{X_t\}$ ($t = 1, 2, \dots, m$). By the relevant formulas in backward cloud generator, calculate expectation E_x , entropy E_n , and hyper entropy H_e as follows:

$$\begin{aligned}E_x &= \text{mean}(x_t), \\ E_n &= \sqrt{\frac{\pi}{2}} * \frac{1}{n} \sum_{t=1}^m |x_t - E_x|, \\ S^2 &= \frac{\sum_{t=1}^m (x_t - E_x)^2}{m-1}, \\ H_e &= \sqrt{S^2 - E_n^2}.\end{aligned}\quad (3)$$

Step 4. Feed E_x , E_n , and H_e obtained in Step 3 into a forward cloud generator. Generate a normal random number E'_{n_t} with E_n being expectation and H_e^2 being variance by $E'_{n_t} = \text{NORM}(E_n, H_e^2)$, and a following normal random number x_t with E_x being expectation and $E'^2_{n_t}$ being variance by $x_t = \text{NORM}(E_x, E'^2_{n_t})$. Then calculate the certainty degree d_t as follows:

$$d_t = \exp \left[-\frac{(x_t - E_x)^2}{2(E'_{n_t})^2} \right]. \quad (4)$$

Repeating these calculations k times will produce the set of cloud droplets $\{x_t, d_t\}$ ($t = 1, 2, \dots, k$), which is denoted by $\{P_t\}$ (the preliminary predication result). Then process $\{P_t\}$ by reversed difference operation twice in combination with $\{\Delta y'_t\}$ and $\{y'_t\}$, and that will yield the final result.

4. The Virtual Machine Migration Strategy

Virtual machine migrations in cloud computing environment are generally composed of the following steps: (1) determine

FIGURE 2: The cloud generators.

FIGURE 3: The flow chart of the time series workload prediction via cloud model.

the source host machine from which some virtual machine is to be removed; (2) determine the destination host machine to which some virtual machine will be moved; (3) choose an appropriate migration scheme (such as static migration, dynamic migration, or mixed migration); (4) carry out the virtual machine migration; and (5) delete the mirror image files associated with the removed virtual machine on the source host machine.

The novelty of our work is to apply the cloud model based time series prediction technique to the migration of virtual machines, namely, to the selections of source host machine, destination host machine, and the virtual machine to be migrated.

4.1. The Strategy Outline. The controlling idea of our virtual machine migration strategy may be characterized as workload-aware migration (WAM). We use the cloud model based time series prediction technique introduced in Section 3 to forecast the future workloads for each host machine. If there exists a host machine whose workload keeps going beyond the upper workload threshold, then this machine will be labeled as a potential source host machine. In a similar manner, if this machine's workload keeps staying between the upper threshold and the lower threshold, then it will be labeled as a potential destination host machine. The flow chart of the proposed virtual machine migration strategy is shown in Figure 4.

Since the workload of each host machine at a data center changes frequently in accordance with the users'

ever-changing service demands, we need to monitor in real time the changes of the workload for all host machines. This constitutes the data source phase in Figure 4. Collected workload data will then be standardized into desired format as the input to the cloud model based prediction engine. The next step is to forecast the future workloads for host machines by the cloud model based prediction engine, and based on the prediction result, to carry out the migration by appropriately selecting the source and destination host machines as well as the virtual machine to be migrated. Finally, the virtual machine mirror image files on the source host machine will be deleted, and the virtual machine on the destination host machine will be started to continue the user services.

4.2. Workload Prediction. The history of the workload of a host machine is formed by its actual past workload values which are collected by the data collection module at data centers. A machine's CPU workload is used as the indicator for the entire workload of that machine since a higher CPU utilization rate suggests a higher consumption of resource of the machine. In the simulation test for our virtual machine migration strategy, data collected from the PlanetLab [32] platform which can be accessed from within the cloud computing simulation software CloudSim [33] are used as the CPU workload simulation. The set of workload history of host machines is denoted by C_h (For the sake of convenience, we list notations and their meanings used in our workload prediction algorithm in Notation and Their Meanings Section).

FIGURE 4: The flow chart of virtual machine migration.

In order to eliminate the virtual machine migrations which are caused by momentary peak workload values, we analyze in advance the possibility of overload or underload of host machines by forecasting the future workload trend for host machines via the cloud model based workload time series prediction technique. The workload upper threshold and lower threshold are denoted by L_u and L_d , respectively. Both L_u and L_d are heuristic values. The three characteristic values E_x , E_n , and H_e of a cloud are produced by a one-dimensional backward cloud generator, which are then fed into a one-dimensional forward cloud generator to produce the prediction of host machine workloads. The set of predicted future workloads for host machine n (index number of host machine) is denoted by C_f .

4.3. The Selections of Source Host Machine, Destination Host Machine, and Virtual Machine

4.3.1. The List of Source Host Machines. For each set C_f of the predicted workload values, compute its average and denote the result as $L_a(n)$. If

$$L_a(n) > L_u \quad (5)$$

holds, then host machine n will be added to the list Q_s of source host machines. Here, note that the chance of the virtual machine migration triggered by a single momentary peak workload value is eliminated by (5).

Also, if the predicted workload value is below the lower threshold, there would be a need to migrate the virtual machines as well. Specifically, if

$$L_a(n) < L_d \quad (6)$$

holds, then host machine n will be added to the list Q_l of underload host machines, which indicates that all virtual machines on this host machine n need to be migrated.

4.3.2. The List of Virtual Machines. There are two cases to be dealt with for the virtual machine selection depending on whether the host machine is overloaded or underloaded. For the case of underloaded host machines, every virtual machine on every host machine in the list Q_l is to be migrated. Hence all virtual machines will be added to the list S_l which records the index number of the host machine on which a virtual machine resides as well as the average running workload value of the virtual machine. The virtual machine


```

Input:  $C_h, L_u, L_d$ .
Output:  $Q_a$ .
Procedure begin
(1)  for each  $n$  in  $C_h$  do
(2)    Use cloud model to predict  $k$  future workloads for host  $n$ , and push results into list
(3)     $C_f$ . Calculate the average of the predicted workloads and push result into list  $L_a$ ;
(4)    if  $(L_a(n) > L_u)$  then
(5)      Add host ID  $n$  into the list  $Q_s$ , select the VM with the largest average history
(6)      workload value on the host  $n$ , and add the VM into queue  $S_o$ ;
(7)    end if
(8)    if  $(L_a(n) < L_d)$  then
(9)      Add host ID  $n$  into list  $Q_l$ , and add all the virtual machines on the host into  $S_l$ ;
(10)   end if
(11)   if  $((L_a(n) < L_u) \&\& (L_a(n) > L_d))$  then
(12)     Add host ID  $n$  into list  $Q_d$ ;
(13)   end if
(14) end for
(15) for each  $L_m(i)$  in  $(S_o \cup S_l)$  do
(16)   for each  $L_a(j)$  in  $Q_d$  do
(17)     if  $(L_m(i) + L_a(j) < L_u)$  then
(18)       Migrate virtual machine  $i$  to host  $j$ , add VM information into list  $Q_a$ ;
(19)     end if
(20)   end for
(21) end for
(22) return  $Q_a$ ;
end Procedure

```

ALGORITHM 1: The virtual machine migration algorithm.

workload values in this list are sorted from large to small. For the case of overloaded host machines, the list Q_s needs to be traversed to select, on each host machine, the virtual machine with the largest average running workload value and to add the selected virtual machine to the list S_o , which, like the list S_l , records the index number of the host machine on which a virtual machine resides as well as the average running workload value of the virtual machine. Again, as for the previous case, the virtual machine workload values in this list are sorted from large to small. Note that in consideration of the migration efficiency, the available virtual machine with the largest workload value should be chosen to be migrated so that the maximum migration benefit tends to be acquired by the minimum number of migrations.

4.3.3. The List of Destination Host Machines. A host machine will be chosen as a candidate for virtual machine migrations if its predicted workload value is between the upper threshold and the lower threshold. That is, if

$$L_d < L_a(n) < L_u \quad (7)$$

holds, then the host machine n will be added to the list Q_d .

4.4. The Migration of Virtual Machines. The actual migration of a virtual machine can be decided by using the previously obtained lists of host machines and virtual machines. If

$$L_m(i) + L_a(j) < L_u \quad (8)$$

holds, where $L_a(j)$ is taken from the list Q_d and $L_m(i)$ is a virtual machine workload value taken from the list S_o ; then the corresponding virtual machine will be migrated to the host machine indexed by j ; subsequently, $L_m(i)$ and $L_a(j)$ will be removed from the lists S_o and Q_d , respectively. The index numbers of the source host machine and destination host machine will be released back to C_h .

For underloaded host machines, all virtual machines will be migrated to the destination host machines by (8), if the list Q_l is not empty. All information about a virtual machine will be saved into the list Q_a after the virtual machine is migrated. The detailed algorithm for virtual machine migration is shown in Algorithm 1.

5. Simulation Results

Simulation experiments are performed to test the effectiveness of our cloud model based time series workload predication algorithm in comparison with the autoregressive (AR) time series workload prediction model [30]. Data collected via the popular cloud computing simulation software CloudSim [33] in conjunction with global network platform PlanetLab [32] are used as input to predict the future workload trend for host machines.

5.1. Raw Data. Table 1 contains the workload data of a physical node over 96 successive and 5-minute-spaced time points collected on the PlanetLab on April 20, 2011. The first 90 entries in Table 1 are used as sample input to the two workload prediction algorithms, and the remaining 6

TABLE 1: Collected raw workload data.

Original workloads collected (Time: T ($i = 1, 2, \dots, 96$), $T_{i+1} - T_i = 5$ min)									
16.0000	12.0000	15.0000	26.0000	41.0000	27.0000	22.0000	16.0000	14.0000	22.0000
28.0000	60.0000	18.0000	32.0000	23.0000	26.0000	19.0000	6.0000	22.0000	43.0000
30.0000	31.0000	21.0000	27.0000	40.0000	61.0000	32.0000	25.0000	27.0000	26.0000
29.0000	18.0000	16.0000	23.0000	24.0000	17.0000	16.0000	50.0000	17.0000	74.0000
51.0000	22.0000	36.0000	23.0000	20.0000	14.0000	70.0000	22.0000	12.0000	20.0000
19.0000	22.0000	36.0000	23.0000	31.0000	24.0000	77.0000	16.0000	22.0000	19.0000
6.0000	20.0000	22.0000	31.0000	24.0000	18.0000	22.0000	10.0000	18.0000	16.0000
16.0000	20.0000	15.0000	18.0000	20.0000	28.0000	10.0000	28.0000	20.0000	16.0000
80.0000	50.0000	23.0000	24.0000	20.0000	27.0000	15.0000	20.0000	28.0000	24.0000
29.0000	22.0000	18.0000	67.0000	16.0000	42.0000

FIGURE 5: The original workloads and workloads after being treated twice with difference operations.

FIGURE 6: The original workloads and workloads after being normalized.

entries are used to check the correctness and accuracy of the prediction results of the algorithms.

5.2. Experiment Metrics. Standard error measurement tools in statistics mean absolute error (MAE) and root mean squared error (RMSE) are used to evaluate the accuracy of the algorithms. If the predicated workload sequence is $\{p_1, p_2, \dots, p_k\}$ and the actual workload sequence is $\{q_1, q_2, \dots, q_k\}$, then MAE and RMSE are given by

$$\begin{aligned} \text{MAE} &= \frac{\sum_{i=1}^k |p_i - q_i|}{k}, \\ \text{RMSE} &= \sqrt{\frac{\sum_{i=1}^k (p_i - q_i)^2}{k}}. \end{aligned} \quad (9)$$

Although formulated differently and having separate usability and appropriateness, both MAE and RMSE provide an indication as to how well the predication results match the actual data. A smaller MAE value or RMSE value means that the prediction results fit the actual data more closely and thus are superior.

5.3. Experiment Results. Due to their unstable and discrete nature, the originally collected workload data are processed by the difference operation and the normalization operation before they are used to generate the predictions. The results of applying these operations to the original data are depicted in Figures 5 and 6, respectively.

The prediction results of our cloud model based algorithm are compared with that of the AR model; details of the comparison are recorded in Table 2 and are contrasted in Figures 7 and 8. We can see that our cloud model based algorithm has lower values than that of AR in terms of both absolute error and error rate. The MAE value and RMSE value of our algorithm and the AR model are, respectively, 2.9310, 3.0334 and 6.5187, 7.3504, which are shown in Table 3 and graphed in Figures 9 and 10. Again, it can be seen that our cloud model based algorithm yields much more accurate prediction results than the AR model, and thus it can select more appropriate host machines and virtual machines in the process of virtual machine migrations.

Finally, the prediction results of our algorithm and the AR model together with the actual workload data are charted in Figure 11 to present a simultaneous visual comparison

TABLE 2: The comparison of cloud model based algorithm with AR model.

Original load value	Predicted load value		Absolute error		Error rate	
	Cloud model	AR model	Cloud model	AR model	Cloud model	AR model
29.0000	25.1528	26.1580	3.8472	2.8420	13.27%	9.80%
22.0000	24.6020	19.1136	2.6020	2.8864	11.83%	13.12%
18.0000	20.3683	24.6645	2.3683	6.6645	13.16%	37.03%
67.0000	64.5862	58.3297	2.4138	8.6703	3.60%	12.94%
16.0000	20.1804	21.4680	4.1804	5.4680	26.13%	34.18%
42.0000	39.8258	29.4192	2.1742	12.5808	5.18%	29.95%

FIGURE 7: Absolute error comparison between cloud model and AR model.

FIGURE 8: Error rate comparison between cloud model and AR model.

of the effectiveness of the algorithms, and the fact that the prediction results of our algorithm approximate the actual data curve well following the sample data curve is shown in Figure 12. Both of these figures are further evidence

FIGURE 9: MAE comparison between cloud model and AR model.

TABLE 3: MAE and RMSE values of cloud model and AR model.

Metrics	Cloud model	AR model
MAE	2.9310	6.5187
RMSE	3.0334	7.3504

supporting the claim that our cloud model based algorithm is more desirable than the AR model.

6. Conclusion

We proposed in this paper a new virtual machine migration strategy which predicts future workloads of host machines by using the forward and backward cloud generators in cloud model, determines source and destination host machines on the basis of the prediction result and by the WAM criterion, and selects the most resource-demanding virtual machine on source host machines to perform the migration. Through the comparison with the peer AR model time series workload prediction technique, we found that our algorithm clearly delivers a more precise workload prediction result than the AR model, and thus it provides more effective support for the selection of host machines in virtual machine migration

FIGURE 10: RMSE comparison between cloud model and AR model.

FIGURE 11: Comparison of predicted data with the original data.

process, reduces the number of virtual machine migrations, and eventually promotes the system to reach a dynamic resource balance improving the resource utilization rate and the virtual service quality.

Notation and Their Meanings

- C_h : The set of workload history of host machines
- L_u : The workload upper threshold
- L_d : The workload lower threshold
- C_f : The set of predicted future workloads for host machines

FIGURE 12: Cloud model prediction results with respect to original data and sample input.

- $L_a(n)$: The average value of predicted future workloads for host machine (indexed by n)
- Q_s : The list of source host machines
- Q_l : The list of low workload host machines
- S_l : The list of to-be-migrated virtual machines on low workload host machines
- S_o : The list of to-be-migrated virtual machines on high workload (overloaded) host machines
- Q_d : The list of destination host machines
- $L_a(j)$: The average value of predicted future workloads for host machine (indexed by j)
- $L_m(i)$: The average value of predicted future workloads for virtual machine (indexed by i)
- Q_a : The list of migrated virtual machines
- L_a : The average value set of predicted future workloads for host machines.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work is supported in part by the following Grants: National Science Foundation of China (Grant no. 61272400), Chongqing Innovative Team Fund for College Development Project (Grant no. KJTD201310), Chongqing Youth Innovative Talent Project (Grant no. cstc2013kjrc-qnrc40004), Ministry of Education of China and China Mobile Research Fund (Grant no. MCM20130351), and Science and Technology on Information Transmission and Dissemination in Communication Networks Laboratory Open Project (Grant no. ITD-U13002/KX132600009).

References

- [1] A. J. Younge, R. Henschel, J. T. Brown, G. von Laszewski, J. Qiu, and G. C. Fox, "Analysis of virtualization technologies for high performance computing environments," in *Proceedings of the IEEE International Conference on Cloud Computing (CLOUD '11)*, pp. 9–16, Washington, DC, USA, July 2011.
- [2] Q. Li, Q.-F. Hao, L.-M. Xiao, and Z.-J. Li, "Adaptive management and multi-objective optimization for virtual machine placement in cloud computing," *Chinese Journal of Computers*, vol. 34, no. 12, pp. 2253–2264, 2011.
- [3] H. B. Mi, H. M. Wang, G. Yin, D. X. Shi, Y. F. Zhou, and L. Yuan, "Resource on-demand reconfiguration method for virtualized data centers," *Journal of Software*, vol. 22, no. 9, pp. 2193–2205, 2011.
- [4] Y. Wen, D. Meng, and J.-F. Zhan, "Adaptive virtualized resource management for application's SLO guarantees," *Journal of Software*, vol. 24, no. 2, pp. 358–377, 2013.
- [5] C. Clark, K. Fraser, S. Hand et al., "Live migration of virtual machines," in *Proceedings of the 2nd Symposium on Networked Systems Design and Implementation*, vol. 2, pp. 273–286, 2005.
- [6] Y. Wu and M. Zhao, "Performance modeling of virtual machine live migration," in *Proceedings of the IEEE 4th International Conference on Cloud Computing (CLOUD '11)*, pp. 492–499, IEEE, Washington, DC, USA, July 2011.
- [7] J. Hu, J. Gu, G. Sun, and T. Zhao, "A scheduling strategy on load balancing of virtual machine resources in cloud computing environment," in *Proceeding of the 3rd International Symposium on Parallel Architectures, Algorithms and Programming (PAAP '10)*, pp. 89–96, Dalian, China, December 2010.
- [8] C. Weng, Q. Liu, L. Yu, and M. Li, "Dynamic adaptive scheduling for virtual machines," in *Proceeding of the 20th ACM International Symposium on High-Performance Parallel and Distributed Computing (HPDC '11)*, pp. 239–250, New York, NY, USA, June 2011.
- [9] M. Mishra, A. Das, P. Kulkarni, and A. Sahoo, "Dynamic resource management using virtual machine migrations," *IEEE Communications Magazine*, vol. 50, no. 9, pp. 34–40, 2012.
- [10] W. Zhao, Z. Wang, and Y. Luo, "Dynamic memory balancing for virtual machines," *ACM SIGOPS Operating Systems Review*, vol. 43, no. 3, pp. 37–47, 2009.
- [11] Y. Shi, X. Jiang, and K. Ye, "An energy-efficient scheme for cloud resource provisioning based on CloudSim," in *Proceedings of the IEEE International Conference on Cluster Computing (CLUSTER '11)*, pp. 595–599, September 2011.
- [12] K. Ye, X. Jiang, D. Huang, J. Chen, and B. Wang, "Live migration of multiple virtual machines with resource reservation in cloud computing environments," in *Proceedings of the IEEE 4th International Conference on Cloud Computing (CLOUD '11)*, pp. 267–274, July 2011.
- [13] X. Chen and J. Zhang, "Virtual machines scheduling algorithm oriented load forecast," in *Proceedings of the International Conference on Network Computing and Information Security (NCIS '11)*, pp. 113–117, May 2011.
- [14] D. Y. Li, K. Di, D. Li, and X. Shi, "Mining association rules with linguistic cloud models," *Journal of Software*, vol. 11, no. 2, pp. 143–158, 2000.
- [15] M. Andreolini, S. Casolari, M. Colajanni, and M. Messori, "Dynamic load management of virtual machines in cloud architectures," in *Cloud Computing*, vol. 34 of *Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering*, pp. 201–214, 2010.
- [16] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. F. de Rose, and R. Buyya, "CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms," *Software: Practice and Experience*, vol. 41, no. 1, pp. 23–50, 2011.
- [17] S. Akoush, R. Sohan, A. Rice, A. W. Moore, and A. Hopper, "Predicting the performance of virtual machine migration," in *Proceeding of the 18th Annual IEEE/ACM International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS '10)*, pp. 37–46, Miami Beach, Fla, USA, August 2010.
- [18] W. Hu, A. Hicks, L. Zhang et al., "A quantitative study of virtual machine live migration," in *Proceedings of the ACM Cloud and Autonomic Computing Conference*, pp. 1–11, 2013.
- [19] V. Medina and J. M. García, "A survey of migration mechanisms of virtual machines," *ACM Computing Surveys*, vol. 46, no. 3, p. 30, 2014.
- [20] Y. C. Chang, R. S. Chang, and F. W. Chuang, "A predictive method for workload forecasting in the cloud environment," in *Advanced Technologies, Embedded and Multimedia for Human-Centric Computing*, vol. 260 of *Lecture Notes in Electrical Engineering*, pp. 577–585, Springer, 2014.
- [21] H. Ren, Y. Lan, and C. Yin, "The load balancing algorithm in cloud computing environment," in *Proceedings of the 2nd International Conference on Computer Science and Network Technology (ICCSNT '12)*, pp. 925–928, Changchun, China, December 2012.
- [22] A. Beloglazov, J. Abawajy, and R. Buyya, "Energy-aware resource allocation heuristics for efficient management of data centers for Cloud computing," *Future Generation Computer Systems*, vol. 28, no. 5, pp. 755–768, 2012.
- [23] A. Beloglazov and R. Buyya, "Energy efficient resource management in virtualized cloud data centers," in *Proceedings of the 10th IEEE/ACM International Symposium on Cluster, Cloud, and Grid Computing (CCGrid '10)*, pp. 826–831, Melbourne, Australia, May 2010.
- [24] A. Beloglazov and R. Buyya, "Adaptive threshold-based approach for energy-efficient consolidation of virtual machines in cloud data centers," in *Proceedings of the 8th International Workshop on Middleware for Grids, Clouds and e-Science*, p. 4, 2010.
- [25] A. Beloglazov and R. Buyya, "Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in Cloud data centers," *Concurrency Computation Practice and Experience*, vol. 24, no. 13, pp. 1397–1420, 2012.
- [26] D. Gmach, J. Rolia, L. Cherkasova, and A. Kemper, "Workload analysis and demand prediction of enterprise data center applications," in *Proceedings of the 10th IEEE International Symposium on Workload Characterization (IISWC '07)*, pp. 171–180, September 2007.
- [27] N. Roy, A. Dubey, and A. Gokhale, "Efficient autoscaling in the cloud using predictive models for workload forecasting," in *Proceedings of the IEEE International Conference on Cloud Computing (CLOUD '11)*, pp. 500–507, Washington, DC, USA, July 2011.
- [28] A. Khan, X. Yan, S. Tao, and N. Anerousis, "Workload characterization and prediction in the cloud: a multiple time series approach," in *Proceedings of the IEEE Network Operations and Management Symposium (NOMS '12)*, pp. 1287–1294, Maui, Hawaii, USA, April 2012.

- [29] Y. Wu, K. Hwang, Y. Yuan, and W. Zheng, "Adaptive workload prediction of grid performance in confidence windows," *IEEE Transactions on Parallel and Distributed Systems*, vol. 21, no. 7, pp. 925–938, 2010.
- [30] Y. Zhao and L. Shen, "Application of time series auto regressive model in price forecast," in *Proceedings of the International Conference on Business Management and Electronic Information (BMEI '11)*, pp. 768–771, May 2011.
- [31] D. Y. Li, "Uncertainty in knowledge representation," *China Engineering Science*, vol. 2, no. 10, pp. 73–79, 2000.
- [32] Princeton University, "PlanetLab: an open platform for developing, deploying, and accessing planetary-scale services," 2014, <https://www.planet-lab.org>.
- [33] Melbourne Clouds Lab, "CloudSim : A Framework For Modeling and Simulation of Cloud Computing Infrastructures and Services," 2014, <http://www.cloudbus.org/cloudsim/>.