Hindawi
Mathematical Problems in Engineering
Volume 2022, Article ID 8190296, 6 pages
https://doi.org/10.1155/2022/8190296

Research Article

Virtual Machine Allocation Strategy Based on Statistical Machine Learning

Bo Han [1,2] and Rongli Zhang [1,2]

[1] *College of Mathematics and Computer Application, Shangluo University, Shangluo 726000, Shaanxi, China*
[2] *Engineering Research Center of Qinling Health Welfare Big Data, Universities of Shaanxi Province, Shangluo 726000, Shaanxi, China*

Correspondence should be addressed to Bo Han; 232021@slxy.edu.cn

Received 19 April 2022; Accepted 11 June 2022; Published 5 July 2022

Academic Editor: Zaoli Yang

At present, big data cloud computing has been widely used in many enterprises, and it serves tens of millions of users. One of the core technologies of big data cloud service is computer virtualization technology. The reasonable allocation of virtual machines on available hosts is of great significance to the performance optimization of cloud computing. We know that with the continuous development of information technology and the increasing number of computer users, different virtualization technologies and the increasing number of virtual machines in the network make the effective allocation of virtualization resources more and more difficult. In order to solve and optimize this problem, we propose a virtual machine allocation algorithm based on statistical machine learning. According to the resource requirements of each virtual machine in cloud service, the corresponding comprehensive performance analysis model is established, and the reasonable virtual machine allocation algorithm description of the host in the resource pool is realized according to the virtualization technology type or mode provided by the model. Experiments show that this method has the advantages of overall performance, load balancing, and supporting different types of virtualization.

1. Introduction

Cloud server, also known as a cloud computing server, is a network server that provides network services for users [1]. Compared with VPS, the cloud server has a great improvement in flexibility, operability, and scalability. Moreover, the rental price is far lower than that of the physical server [2]. It is very convenient for users to open and use when using. Moreover, in the late need to expand the use of resources, you can also quickly upgrade the server configuration [3]. Therefore, the design of many enterprise data centers is based on cloud computing service mode. This technical mode requires all hardware resources to be concentrated in a common basic resource pool that everyone can share [4]. When they need to be applied, these resources can be used through client applications. This forms the so-called virtual service environment in our computer industry. Virtual service environment allows user program and data information to be carried out among different types of virtual machines [5]. Data center operators prepare the storage of virtualized resources in the back end according to the needs of customers and provide them in the form of a storage resource pool. Customers can use this storage resource pool to store files or objects. In short, cloud storage is an emerging solution to put storage resources on the cloud for people to access. Users can connect to the cloud at anytime and anywhere through any network connected device to access data conveniently. Generally, there are thousands of virtual machines active in the network [6]. The question of how to efficiently allocate thousands of virtual machines among the hosts in the resource pool according to the types of virtualization services requires a reasonable and efficient virtual machine allocation strategy because it can dynamically allocate virtual machines, integrate the workload of servers, and allocate required resources, which can effectively improve the network server and maintain the required quality of service and improve the utilization of physical resources. For different types of virtualization and heterogeneous hosts, the relationship between virtual machines and physical hosts is different. Accurately capturing the relationship between virtual machine and host has a great impact on the allocation of virtual machine [7, 8]. The commonly used performance models, such as linear model and simple queuing model, are often unrealistic and cannot adapt to the change of load [9].

In order to solve the above problems, we propose a workload estimation model based on statistical machine learning (SML) modelling, control, and analysis technology combined with a target host selection algorithm.

In order to estimate the resource requirements of virtual machines, we use the SML method to describe a workload estimation model, which predicts the resource set required by each virtual machine. We also provide an overview of the effective utilization of these resources. Then, aiming at the problem of load balancing between hosts, a virtual machine allocation algorithm is proposed. The advantage of this strategy is that when the virtual machine uses different virtualization types, it can choose a better server for the virtual machine to meet the requirements of load balancing.

2. Related Work

As the trend of the future computing model and the core of the new generation of information technology and business model transformation, cloud computing has attracted more and more attention from researchers and enterprises and has broad market development prospects [10]. At present, almost all its giants are marching into cloud computing from different directions according to their technical advantages and market strategies. At present, cloud computing application services are becoming more and more popular, the scale of data center supporting cloud computing is becoming larger and larger. The data center forms a huge virtual resource pool by using virtualization technology [11]. However, due to the lack of effective resource management mechanism, virtual machine resources cannot be reasonably allocated. For example, with the operation of the system and the change of user service load, the placement of virtual machine will become disorderly. In addition, if we can regularly plan and deploy the virtual machine from the global scope, it can also improve the resource utilization of the data center to a certain extent. At present, the existing algorithms and research projects, such as VMware DRS and IBM virtualization manager, dynamically allocate resources to the partition/virtual machine according to the static specified sharing and resource utilization rate, while ignoring the quality of service [12]. In this paper, after a thorough discussion and detailed study of the existing algorithms, this paper proposes a method like VMware-DRS, which uses the performance parameters of virtual machine as a host utilization function. In order to make the virtual machine dynamically match and meet the requirements of automatic adaptation, the algorithm also adopts the common active control theory and describes the process of virtual machine call and resource allocation as a feedback control problem, so that it can be placed and adjusted in the server. In order to optimize the resource performance and balance the virtual machine load power cost management.

Compared with the existing technologies, the main disadvantage of both VMware DRS and IBM virtualization manager is that these algorithm models rely on the cost function to determine how to respond to the workload input parameters. What is the cost function? Cost function is a user-defined rule paradigm, which needs to be constructed by computer-related domain knowledge and usually varies according to the size and variability of workload and system subcomponents. Unlike them, our strategy uses virtual machines as a vehicle for building applications. By using application load and performance prediction technology, we can get a more general virtual machine function loading set and realize online resource estimation and virtual machine allocation algorithm of the virtual data center through SML. The algorithm uses the available resource consumption and performance information in the system to learn and predict the workload.

3. The Allocation Strategy

Using the abovementioned algorithm, we get a virtual machine allocation model and strategy method, which can dynamically adjust the resource allocation of data center through the algorithm parameters and the changing workload of virtual machine. The strategy is executed as follows:

Step 1. The historical data recorded on the sampling server (1 day) are used to predict the workload of the performance model in the next hour with the nonlinear model KCCA. It helps to describe the trend of messy data and the overall situation of large data sets. It needs the resources needed to meet the following workload.

Step 2. Our virtual machine allocation algorithm based on load balancing uses the traditional correlation analysis KCCA algorithm to capture the interdependence between CPU and I/O resource indicators of different hosts and uses this relationship to predict the resource consumption and host load information of the virtual machine. Finally, through this interdependence relationship, we select the related hosts in the resource pool according to the relevant performance parameters the machine is paired.

Step 3. First, each host in the resource pool is assigned a relationship parameter a to find the host with the smallest deviation in the resource pool.

Step 4. KCCA method and RRDs are used to capture CPU, memory, and I/O data information of different hosts.

Step 5. The Erne function is used to establish the nonlinear relationship. Here, we will focus on why we choose to use the Erne function to model the nonlinear relationship? We know that the technical principle of virtualization determines that the change of virtual machine resource utilization may lead to the change of

```
Data symbol description:
$R$: Resource pool.
VH: Obtained virtual machine data.
50: Host.
$C$: Virtual machine.
$N$: Number of virtual machines.
$T_h$: Target host.
$P_l$: Load of the virtual machine.
Input parameters:
Virtual machine data information (VH) already exists in the resource pool ($R$). Assuming that there are many ($n$) active virtual host information in the resource pool, the configuration information of a virtual machine ($c$) can be used by a host ($L$) for assembly scheduling or resource utilization.
Output:
The information of the target host in the resource pool is $T_h$.
The virtual machine scheduling type is $T_{n.}$
Using KCCA algorithm to estimate the load information of virtual machine $p_{i.}$
For each host do
Load$_i = p_i$
Use (1) to compute the host prating $P_i$ and $F_i$
End for
$T_h = C_i$//initialize the target host
While $i \leq n$ do
If $L_i < C$//judge if it meets the configuration requirements
There is no suitable host
Else
$T_h = i$//virtual machine is allocated on host i
If Ty is para-virtualization
Load$_i$ = load$_i + F_i{}^*\rho$//the load of the VM is added to host i
Avgload$_i$ = $\sum_{i=1}^{n}$ load$_i$/$n$//compute the average load of the resource pool $R$
$d_i = \sqrt{\sum_i^n (\text{load}_i - \text{avgload}_i)^2/(n-1)}$ //compute the deviation of the load
Load$_i$ = load$_i$-F$_i{}^*\rho$
End if
$i = i + 1$
End if
End while
Find the minimization of the $d_i$
$T_n = i$//the virtual machine is allocated on host
```

Algorithm 1: The virtual machine allocation algorithm.

host resource utilization. Therefore, we need to capture the association between virtual machine and actual host, which is the similarity relationship between them. The Erne function provides higher efficiency and expressiveness in obtaining the similarity of objects. The most important thing is that it can use its correlation to quantify the performance similarity of objects, which is incomparable to other functions.

According to the abovementioned strategy, the relationship between virtual machine and physical host is established by multiple regression analysis. The formula is as follows:

$$\text{Hi} = a_1 + a_2 * v(\text{cpu} + a_3 * v(\text{menu}) + a_4(IO)). \tag{1}$$

The parameters are described as follows:

H: It represents the percentage of virtual machine usage in total resource capacity during the test, i.e., host utilization.

V: It represents the percentage of the total resource capacity used by the virtual machine during the test, that is, the resource utilization of the virtual machine.

H_i: It represents the load of a host I in the resource pool during the test.

V (CPU): It represents the CPU utilization of the virtual machine per minute during the test.

A: It represents the percentage of virtual machine usage in total resource capacity.

After processing the data relationship, we can use two relations to express the virtualization degree of the server and a virtual machine. The value of the virtualization degree is the service level of the server. When the server receives the service request instruction from the virtual machine, the service center will calculate the virtual machine resource utilization VI to the server according to the server level and the workload H_i of the virtual machine and calculate the CPU, network I/O, and other indicators of the virtual machine, so as to guide us to give a better virtual machine layout strategy, so as to make the total load balance.

4. Experimental Results

In this section, we will experiment to show that our method can effectively allocate virtual machines. First, we prove the capability and accuracy of our relationship model in response to virtual machine resource requests. Then, we prove that the algorithm is effective for selecting the optimal host for load balancing. Therefore, in the experiment, we need to use two identical servers and a storage array for recording resource information. They are interconnected by high-performance optical fibers to form a LAN and finally form a shared service resource pool. All the virtual machines in our experiment are cloned from two virtual machine templates, one of which is quasi virtualization and the other is full virtualization. This design is to ensure that the dependency relationship between them is as concise and clear as possible.

$$\text{ERMSE} = \left(\frac{1}{N-1}\sum_{t-1}^{n}\left[p(t)-O(t)\right]^{2}\right)^{1/2}, \tag{2}$$

$$\text{Epa} = \frac{\sum_{t-0}^{n}\left[\left(P(t)-P_{m\left(O(t)-O_{m}\right)}\right)\right]}{(N-1)\sigma_{p}\sigma_{o}}. \tag{3}$$

In the experiment, ERMSE (2) and Epa (3) were used to evaluate the accuracy of the prediction results. Therefore, we assume that the predicted value is P(T), the actual value is O(T), the standard deviation is σ_{P}, and the actual error is σ_{o}. The experimental results show that the ERMSE value reflects the average deviation between the predicted value and the actual average value, and its value close to zero means that the prediction is close to perfect. The experimental results show that the EPA value reflects the correlation between the deviation between the predicted value and the mean value and between the actual value and the mean value. Its value is between 1 and −1, and Table 1 indicates that the prediction result is almost perfect.

In the experiment, we use the KCCA method to test our algorithm and compare the results with the existing regression method. The comparison results are shown in Table 1. The experimental results show that KCCA is more ideal than the regression method. Therefore, our algorithm has significantly improved the accuracy of virtual machine prediction, so it has a better prediction effect.

We use 12 virtual machines and two servers for testing. We choose load balancing as the evaluation criteria. When using this algorithm, the system will choose the host for virtual machine while starting according to the characteristics of virtual machine's load and the host rating. First, we use this rating to calculate the host resource utilization and compare it with the actual results. The results demonstrate that it gives a better solution when the virtual machines use different virtualization technologies.

Table 2 shows the residuals and coefficient of calculation for different type of virtualization. Also, we compared this strategy with the greedy algorithm which always chooses the node that has the lowest load. Figure 1 shows the CPU utilization using the simple algorithm when placing the 12 virtual machines on these two hosts. Figures 2(a) and 2(b) gives the CPU utilization of using our strategy. It can be clearly seen from (a) and (b) that the CPU load was not balance, for the load on host 1 is 0.1 but the load on host 2 is about 0.3. However, Figure 2 shows that when using our strategy, the load of these two hosts is load balancing for the average load is almost the same. Experimental results show that our strategy can effectively guide the allocation of virtual machines to achieve the load balancing.

Table 1: Comparison of the accuracy of KCCA and regression.

Metric	KCCA		Regression	
	ERMSE	Epa	ERMSE	Epa
B_{i}	0.12	0.88	0.47	0.69
B_{o}	0.13	0.88	0.44	0.66
C_{s}	0.096	0.87	0.21	0.66
R (kb/s)	0.095	0.87	0.14	0.75
W (kb/s)	0.097	0.85	0.13	0.64

Table 2: The residual and coefficient of the strategy.

Virtualization	CPU		Network I/O	
Type	Res	Coe	Res	Coe
Para-virtualization	0.074	0.99	0.0019	0.99
Full virtualization	0.0014	0.97	0.0029	0.96

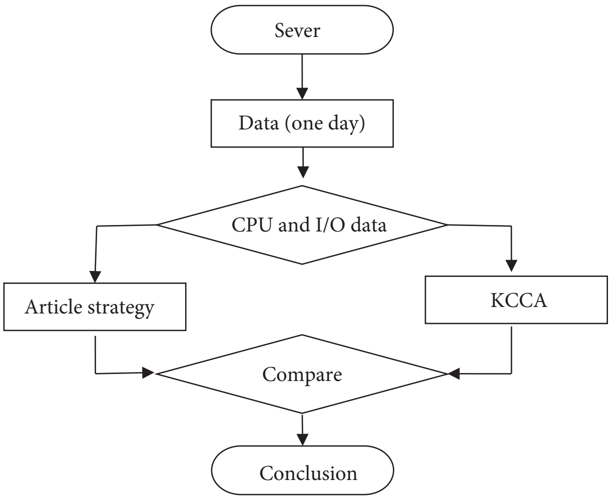
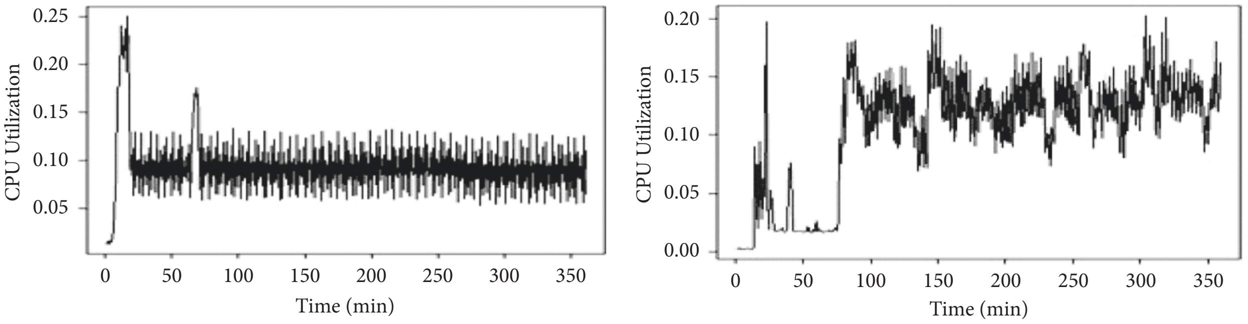


Figure 1: Strategy flow chart.

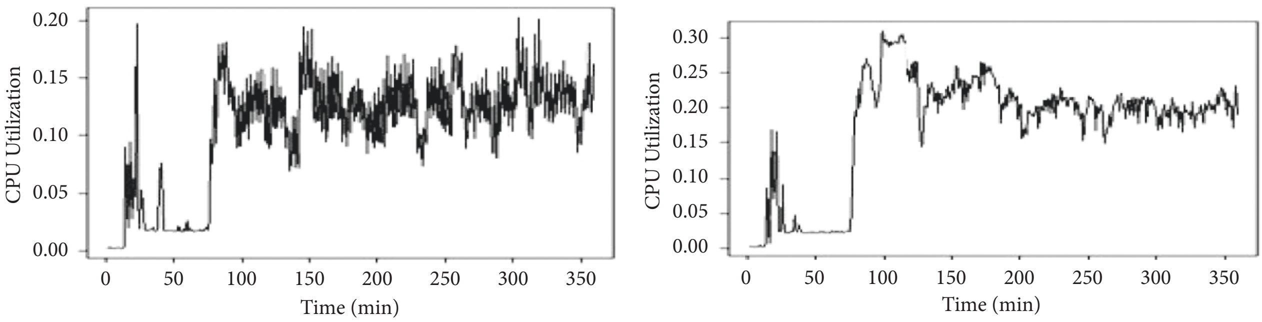
Table 2 shows the residuals and coefficient of calculation for different types of virtualization. Also, we compared this strategy with the greedy algorithm which always chooses the node that has the lowest load. Figures 2(a) and 2(b) show the CPU utilization using the simple algorithm when placing the 12 virtual machines on these two hosts, Figure 3(a) and 3(b) give the CPU utilization of using our strategy. It can be clearly seen from (a) and (b) that the CPU load was not balance, for the load on host 1 is 0.1 but the load on host 2 is about 0.3. But Figure 3 shows that when using our strategy,



(a)

(b)

Figure 2: (a) CPU utilization of two hosts using the greedy algorithm. (b) CPU utilization of two hosts using the greedy algorithm.



(a)

(b)

Figure 3: (a) CPU utilization of two hosts using our algorithm. (b) CPU utilization of two hosts using our algorithm.

the loads of these two hosts are almost the same under the same response time. Experimental results show that our strategy can effectively guide the allocation of virtual machines to achieve the load balancing.

Next, we test the superiority of our algorithm in the evaluation of virtual machine load balancing. We used more (30) virtual machines and our servers for testing. When using the algorithm, the system will select the host for the virtual machine according to the characteristics of the virtual machine load and the host rating and calculate the host resource utilization. The experimental results in Table 2 show that compared with the greedy algorithm, the greedy algorithm always selects the host with the lowest load, while our algorithm performs better in CPU utilization of virtual machine and network I/O and other aspects of the load is balanced; the algorithm can effectively guide the allocation of virtual machines, achieve more effective load balancing, especially when the virtual machine adopts different virtualization technology; the method is more reliable, more reasonable, and feasible.

5. Conclusions

We propose a virtual machine allocation algorithm based on statistical machine learning. The strategy is to establish the corresponding comprehensive performance analysis model according to the resource requirements of each virtual machine in the resource pool. According to the virtualization technology type or mode provided by the model, the reasonable virtual machine allocation algorithm description of the host in the resource pool is realized. Experimental results show that our method is more ideal than the regression method in virtual machine allocation prediction results, and it is easier to achieve host load balancing than the greedy algorithm. Therefore, compared with other existing methods, this algorithm has the advantages of good overall performance, load balancing, and supporting different types of virtualization.

Data Availability

The data supporting the findings of this study are available within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by Shaanxi province 2016-2017 basic education major bidding project, the level of students' artistic education in Shaanxi Province "Research on the Measurement System" (ZDKT1606). This work was supported by Shaanxi education society 2016 project of the year, "Research on Strategies for Promoting Adolescent Health

Growth under the Network Environment" (SJHZDKT2016). This work was supported by Teaching reform project of education in Shangluo college, "Research on "flipped classroom" Teaching Mode of "Software Engineering" under "Internet +" Era" (17jyjx011). This research is supported by the Educational science planning project of Shaanxi Province "Research on Flipped Classroom Teaching Mode of Software Development Courses in the Era of Internet + ".

References

[1] J. Waring and C. Lindvall, "Review of the state-of-the-art and opportunities for healthcare," *Renato Umeton Automated machine learning*, vol. 16, no. 12, p. 104, 2020.

[2] E. S. Carl and L. Darya, "Zabelina Classifying creativity,"Applying machine learning techniques to divergent thinking EEG data"," *NeuroImage*, vol. 8, no. 6, p. 219, 2020.

[3] N. Tamascelli and N. Paltrinieri, "Valerio Cozzani Predicting Chattering Alarms, "A machine learning approach"," *Computers & Chemical Engineering*, vol. 3, no. 2, p. 39, 2020.

[4] R. Chowdhury, M. A. Rahman, M. S. Rahman, and M. Mahdy, "An approach to predict and forecast the price of constituents and index of cryptocurrency using machine learning," *Physica A: Statistical Mechanics and Its Applications*, vol. 551, no. 4, Article ID 124569, 2020.

[5] J. Sampedro, "Machine learning to predict stent restenosis based on daily demographic," *Clinical, and Angiographic Characteristics*, vol. 36, no. 10, p. 34, 2020.

[6] G. Moon, C. Jong-ryul, C. Lee, O. Youngjin, H. K. Kyung, and K. Donghyun, "Machine learning-based design of metaplasmonic biosensors with negative index metamaterials," *Biosensors and Bioelectronics*, vol. 25, no. 6, p. 164, 2020.

[7] E. Moe, B. Mu, K. H. G. Francesca et al., "A novel machine learning approach for predicting the 3D printability of medicines," *International Journal of Pharmaceutics*, vol. 12, no. 12, p. 590, 2020.

[8] G. D. Barmparis, G. Neofotistos, M. Hizanidis, J. Tsironis, G. Kaxiras, and E. Kaxiras, "Robust prediction of complex spatiotemporal states through machine learning with sparse sensing," *Physics Letters A*, vol. 384, no. 15, Article ID 126300, 2020.

[9] R. Feng, "Improving uncertainty analysis in well log classification by machine learning with a scaling algorithm," *Journal of Petroleum Science and Engineering*, vol. 196, no. 4, Article ID 107995, 2021.

[10] G. Qian, Y. Zhou, C. Li, X. Feng, and D. Feng, "Power allocation scheme based on support vector machine forDAS and CAS," *Physical Communication*, vol. 38, no. 12, Article ID 100941, 2020.

[11] W. U. Khan, F. Jameel, M. A. Jamshed, H. Pervaiz, S. Khan, and J. Liu, "Efficient power allocation for NOMA-enabled IoT networks in 6G era," *Physical Communication*, vol. 39, no. 13, p. 53, Article ID 101043, 2020.

[12] Z. Zhang, X. Limin, L. Yongnan, and R. Li, "A VM-based resource management method using statistics," in *Proceedings of the 18th International Conference on Parallel and Distributed Systems*, vol. 6, no. 17, p. 788, December 2012.