IJNRD.ORG ISSN : 2456-4184
INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD) | IJNRD.ORG
An International Open Access, Peer-reviewed, Refereed Journal

A Virtual Machine Scheduling Strategy based on Workload Prediction and Simulated Annealing Algorithm in Cloud Computing

[1]Hanwu Wang, [2]Evangeline Tuyishimire,

[1]Department of Information Science and Technology, Faculty of Computer Engineering,
[1]Hunan University, Changsha, Hunan,China

[2]Department of Information Science and Engineering, Faculty of Computer Science and Technology,
Hunan University, Changsha Hunan, China

Abstract : Cloud computing is a system that helps customers manage IT infrastructure. However, the energy consumption of cloud data centers remains a problem. There are many ways to tackle this problem. Virtualization technology provides an effective solution for the efficient use of data center resources. It allows cloud service providers to run multiple virtual machines on a single server at the same time. Online migration technology can realize the dynamic integration of virtual machines to a small number of servers that satisfy resource requests. The deployment of virtual machines based on virtualization technology has become a research focus in the world. Currently, the initial deployment of virtual machines is mainly based on the degree of performance matching. We consider two types of virtualization and issues of both aspect. For one, the lack of consideration of the type of virtual machine load makes it impossible to efficiently use server resources and also leads to resource competition. On the other hand, the second and current dynamic deployment of virtual machines does not take into account the changing trend of server load, and cannot meet the dynamically changing cloud computing environment. For the above two issues, the specific work of this paper is as follows:
(1) For the initial deployment of virtual machines, a virtual machine allocation method based on load type awareness is proposed. This method aims at energy consumption optimization and load balancing. At the same time, it considers four types of resources: CPU, disk, network bandwidth, and memory requirements. Minimizes the deployment of virtual machines that consume the same type of resources to the same server. Experimental results show that the proposed algorithm effectively reduces energy consumption.
(2) For the dynamic deployment of virtual machines, an efficient scheduling method for virtual machines based on load forecasting is proposed. First use the time series model ARMA to predict the change of server load in advance, and determine the migration timing of the virtual machine through the delay mechanism to avoid frequent migration of the virtual machine. Second, use a simulated annealing algorithm to find a suitable destination server for the virtual machine to be placed. Experimental results show that the proposed algorithm can sharply reduce the number of virtual machine migrations and significantly reduce energy consumption.

IndexTerms **- Cloud Computing, Live Migration; Workload Prediction; SA Algorithm.**

1. INTRODUCTION

Cloud computing moves the IT infrastructure from the local to the cloud resource pool and manages them in a unified way. One way is utilizing virtualization technology to allocate computing resources to end users in the form of on-demand. It does this, by allocating availability, reliability, scalability, and security to the appropriate party. Given the development of cloud computing, the number of hosts in the pool is increasing accordingly. To efficiently use host resources to ensure the quality of service while reducing the total energy consumption is a challenging issue in cloud computing [1]. The most common and effective way to achieve efficiency in cloud resources is to carry on performing the live migration of virtual machines [2]. In practical applications, more hosts must be kept open to meet the user demand and ensure the quality of service. On the other hand, to efficiently energize saving, VM (virtual machine) migration is employed to place the VMs on a small numbers set of host nodes, and turn off those hosts which are in an idle state. However, this will cause frequent scheduling for host nodes and migration of virtual machines, which will result in additional energy consumption and make the cloud computing system unstable.

To solve the above problems, this paper proposes a virtual machine dynamic scheduling algorithm based on workload prediction and simulated annealing (WPSA). First, because the workload information of a virtual machine is a time series, this paper

uses the time series prediction model (ARMA) to predict the workload of the virtual machine in the cloud computing system. ARMA can grasp the workload change of the next time in advance according to the historical workload information, so it can successfully avoid frequent virtual machine migration and achieve resource reservation. Second, in order to save energy consumption, the simulated annealing algorithm is applied to assign the virtual machine that needs to be migrated to the hosts with the least energy consumption increment according to the prediction results. The experimental results show that the proposed algorithm can effectively reduce the number of virtual machine migrations and reduce the total energy consumption of the system.WPSA algorithm. At last we will talk about the performance evaluation and future works of what is to follow.

2. RELATED WORKS

Aiming at given the problem of high energy consumption in cloud computing data centers, several methods for energy awareness and load balancing problems have been put forward [3][4][5]. The authors have performed a number of experiments to obtain the relationship between interaction and energy [6]. Author designed a new energy virtual awareness machine integration heuristic framework to improve power consumption while maintaining the quality of service. The concentrated on VM consolidation to reduce energy consumption and improve resource utilization [7]. And put the problem into two sub-problems: first, finding an overloaded host, and second, using a multidimensional decision-making method to find a suitable host. The approach proposed dynamically merges hosts through VM consolidation and uses a fixed threshold to limit the maximum utilization of resources [8]. They observe the key performance indicators of VMs and merged the servers according to the observed values. If the resource exceeds the predefined threshold, in this case, the push method starts to work and migrate VM to another host. On the other hand, the pull method reallocates resources when the host's workload is low. However, there may be SLA violations due to variable workload. The proposed a heuristic binary search algorithm that tries to place VMs in a few hosts to reduce the energy consumption of the data center [9]. It puts VMs of the same users in the same host to reduce network consumption. This algorithm can't be used very well because it only considers a resource type.

The proposed a method to deal with the dynamic migration of VMs by dynamically adjusting the CPU utilization threshold, and the CPU utilization of virtual machines is predicted by historical records [10]. The authors build an accurate prediction model which uses the distributed decision support system (DDSS) to predict the CPU utilization [11]. The approach proposed focuses on the prediction of the workload of the server [12]. They find the size of an optimal prediction window through experiments and apply it to the migration process of the virtual machine, which optimizes the energy consumption of the whole system The authors use the curve fitting prediction (CFP) technology and genetic algorithm (GAs) to optimize the parameters of the Gauss prediction model [13]. The results show that the algorithm provides a very accurate workload forecast, but the cost of prediction is too high.

3. MODEL and FORMULATION

In this part, we first describe the main features of VMs and hosts based on the multidimensional resource model, then we introduce some concepts and the energy consumption model.

3.1. Problem Definition

We define the set of virtual machines as VM= $\{v_1,v_2,\ldots,v_m\}$, where m is the number of VM types, the VM in each type is modeled by $v_i=\{v_i^{CPU}, v_i^{mem}, v_i^{bw}\}$, which specifically represents the CPU, disk memory and network bandwidth of the VM. Suppose that there are n hosts in the cloud computing resource pool. Each host is characterized by the form of $h_j= \{h_j^{cpu}, h_j^{mem}, h_j^{bw}, P_j^{static}, P_j^{dynamic}, S\}$, where j is a unique identifier of a host, h_j^{cpu}, h_j^{mem}, h_j^{bw} represent the CPU, disk memory and network bandwidth of the host. P_j^{static} refers to the power on the server without running the VM and $P_j^{dynamic}$ refers to the execution power of the $host_j$. Because of the heterogeneity of hosts, the power of VM on different hosts is different, so $P_j^{dynamic}$ is formed of $P_j^{dynamic}= (P_{1j}, P_{2j},\ldots, P_{ij})$, where P_{ij} indicates the power of v_i executing on h_j. S is the bit vector and is formed of S= $\{s_{i1}, s_{i2},\ldots, s_{ij}\}$. If v_i is allocated to the h_j, s_{ij} is set to 1, otherwise, the value of s_{ij} is 0.

3.2 Energy Consumption Model

In order to find the influence of all kinds of resources on energy consumption, the authors in [14] measured the power of hosts by a large number of real experiments and the collected mass data contains power and CPU utilization. Finally, they found that the power of the host is linearly related to the utilization of the CPU. So, in this paper, we use formula (1) to calculate the power of the host.

$$P(u) = P_{static} + P_{dynamic} * u \qquad (1)$$

Where, u refers to the CPU utilization of the host, the calculation formula is as follows:

$$U_{cpu} = \frac{\sum_{i=1}^{n} V_i^{cpu}}{h^{cpu}} \qquad (2)$$

Where, v_i^{CPU} is the CPU usage of v_i, and h^{CPU} is the CPU value assigned to the host. Due to the CPU utilization varying with time, the energy consumption of the host should be a function of time: u (t), so the total energy consumption of the host from t to t+Δt is:

$$E = \int_t^{t+\Delta t} P(u(t))dt \qquad (3)$$

So, the objective in this paper is:

$$\min \sum_{i=1}^{n} E_i \tag{4}$$

Constraints:

$$\sum_{i=1}^{m} s_{ij} \cdot v_i^{cpu} \le h_j^{cpu}, 1 \le j \le n \tag{5}$$

$$\sum_{i=1}^{m} s_{ij} \cdot v_i^{mem} \le h_j^{mem}, 1 \le j \le n \tag{6}$$

$$\sum_{i=1}^{m} s_{ij} \cdot v_i^{bw} \le h_j^{bw}, 1 \le j \le n \tag{7}$$

$$\sum_{j=1}^{n} s_{ij} = 1, 1 \le i \le m \tag{8}$$

Where, the (5) - (7) indicate that the resource request of all virtual machines deployed on the server can't exceed the total amount of resources allocated on the server, and (8) means that any VM can only be assigned to a server for processing

4. DYNAMIC SCHEDULING ALGORITHM

4.1. System Architecture

The cloud computing resource scheduling system process is depicted in Figure 1, including resource monitor, workload predictor, VM scheduler, and resource manager.

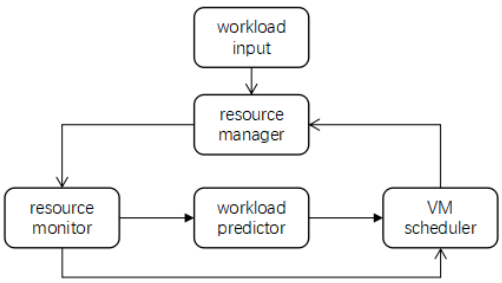

Figure 1: Scheduling System

The resource monitor module is responsible for monitoring the real-time load information of the VMs and hosts in the cloud data center. The load predictor module analyzes the historical data of the current server and predicts the load change of the load server as accurate as possible to determine whether the server is in a state of high load or low load. The virtual machine scheduler module uses the simulated annealing algorithm to reallocate the virtual machine to the appropriate server based on the virtual machine's load and prediction results. The resource manager is mainly responsible for the intelligent switch machine of the server. When the server is in an idle state, the resource manager will turn off the host.The most important feature of the model is that the future workload can be grasped in advance by load forecasting, so the VM scheduler can comfortably perform the virtual machine migration, and the SA algorithm can select the lower energy consumption destination host according to the current state.

4.2. Workload Prediction

Workload affects the efficiency of task execution and has a significant effect on energy consumption in the cloud environment. According to the self-similarity of the workload changes and the strong correlation with time, we can predict future workload values based on historical data values. In order to grasp the load information in real time, we use the ARMA technique for the prediction of CPU utilization. The ARMA technique uses the time-series based forecasting technique and is widely applied in practical time series analysis systems.

Suppose we want to predict the next value of time series $\{X_n\}$ and the previous n-1 values are known. The prediction value X_n is formulated as follows:

$$X_n = \alpha_1 X_{n-1} + \ldots + \alpha_p X_{n-p} + \\ + \varepsilon_n - \beta_1 \varepsilon_{n-1} - \ldots - \beta_q \varepsilon_{n-q} \tag{9}$$

where α is the parameter of autoregressive (AR) part, β is the parameter of moving average (MA) part, and ε is the error term which is generally regarded as Gaussian noise, namely,

$$E(\varepsilon_n)=0, E(\varepsilon_n,\varepsilon_{n+k})=\begin{cases}\delta_\varepsilon^2, k=0\\ 0, k\neq 0\end{cases} \quad (10)$$

In our work, $\{X_n\}$ is a set of history CPU utilization records of a VM, including the most recent data values.

There are two points of concern related to the ARMA technique. The first point, the choice of the model order is considered to be a good practice. Increasing the value of p and q is helpful to obtain better predictive value, but it will lead to more estimated parameters which will make the model too sensitive to data and reduce the robustness of the model. Meanwhile, the smaller the P and Q values may make the computation smaller. However, the prediction accuracy is also lower. In order to find the best model based on the actual sequence, according to the Pandit-Wu method, the order can be made in the way of ARMA (2n, 2n-1). Finally, we choose the values of p and q that minimize the AIC (Akaike Information Criterion) value as the order of the final workload prediction model.

The second point, the choice of model parameters has a great influence on prediction. We use the least squares regression method to get the appropriate values of the parameters. We define the quadratic sum of ε as S, which is formed as

$$S=\sum_{n=p+1}^{N}\varepsilon_n^2 \quad (11)$$

Where,

$$\varepsilon_n=\begin{cases}0, n\le p\\ X_n-\sum_{i=1}^{p}\alpha_i X_{n-i}+\sum_{j=1}^{q}\beta_j\varepsilon_{n-j}, n=p+1,;;N\end{cases} \quad (12)$$

Now, we set the appropriate values which make S get the minimum value as the parameters of the ARMA model.

4.3. VM Migration Policies

The live migration of virtual machines mainly includes two parts: (1) Select VMs for migration to optimize the resource allocation; and (2) select the destination host to place the required VM*s*.

a. VM selection

In order to reduce energy consumption and ensure computation efficiency, we define the upper and lower thresholds of the host CPU utilization respectively.

The lower threshold is mainly concerned with energy consumption. If the host CPU utilization is lower than the lower threshold, it is necessary to move all VMs from this host to make the host idle and turn off the host to eliminate power consumption directly. The upper threshold is to lighten the load of a high workload host and ensure computation efficiency.

When there is a high workload host, some VMs should be chosen and be migrated to prevent SLA violation. Before selecting the VM migration list, we sort the VM list in the descending order of future CPU utilization based on the predicted results, and then choose the previous VM, in turn, to migrate until the remaining resources on the host can meet the future resources of the remaining VMs. The advantage is that it can reduce the number of virtual machine migrations and reduce SLA violations.

The purpose of virtual machine selection is to select virtual machines from under-loaded or overloaded servers, to migrate to other nodes to reduce energy consumption, and SLA violation rate. We define the upper and lower thresholds of server CPU utilization respectively. Among them, the lower threshold TL is mainly considered to reduce energy consumption, and the upper threshold TH is defined to reduce the load on high-load servers and reduce the SLA violation rate. As shown in formula (13), H represents the load variation law of the server CPU, and f refers to the predicted result of the load. When the server's CPU utilization is below TL, it can be considered that the server is in a low load state at the moment, and H is set to -1; when the server's CPU utilization is higher than TH, it can be considered that the server is in a high load state at this moment, and Set H to 1; if the server's CPU utilization is in between, the server is considered healthy and H is set to 0.

$$H=\begin{cases}-1, f\le TL\\ 0, TL<f<TH\\ 1, f\ge TH\end{cases} \quad (13)$$

In order to avoid sudden changes in server load, this paper uses a delayed trigger mechanism to determine the final load state of the server. Assuming that the length of the collected historical data is N, and the number of times the upper or lower threshold is continuously exceeded is n, set a threshold p, (0,1) p∈. When p<n/N, the server is considered to be under high or low load.

Once the server is considered to be in an out-of-bounds state, high or low load, appropriate selection of virtual machines is required. If a server is under low load, all virtual machines in that server will be removed from that server, and the server will be shut down to directly reduce power consumption. When there are high-load servers in the cloud computing resource pool, some virtual machines on the servers will be selected for migration to reduce the SLA violation rate. In order to reduce the migration loss of virtual machines, first sort the virtual machines on the server in descending order of CPU utilization, and select the virtual machine with the highest utilization rate for migration. Virtual machines until the server is under normal load state.

b. VM Placement

If the VM is migrated to an unsuitable host, it will not only affect the execution efficiency but also increase the energy consumption. To solve the VM placement problem, we use the Simulated Annealing (SA) algorithm which has been successfully applied to solve combination optimization problems [15] [16].

The SA algorithm is inspired by the process of solid cooling annealing in metallurgy. This process starts from a higher initial temperature, and the energy decreases with the decrease of temperature. When the temperature reaches the lowest point, the system energy is minimized. In the resource scheduling of cloud computing, the objective function is formula (4), and the solution space is the set of virtual machine deployment strategies. The process of simulated annealing can be described as the process of finding the minimum of the objective function randomly in the solution space with the characteristics of probability mutation. The advantage of this method is allowed to accept the non-optimal solution with a certain probability and the locally optimal solution is avoided effectively.

In the attempt to find a better new solution, the VM allocation map is not random but based on the execution power of the VM on the host. First, we get the host that satisfies the migration condition and the power after the VM is allocated to the host. Then calculate the power difference between the current power and the power after allocation. The host with the largest power difference is chosen to be the final allocation host.

Algorithm 1 Allocation Algorithm

```
Input: hostList,VMList,
Output: allocationMap of Vms
Sort the VMList in descending order based on the future workload
for each VM in VMList do
  for each host in hostList do
      if host.isSuitablefor(VM) then
         According to formula(3)Calculate powerDiiff (newPower after allocation)
         powerDiff=newPower-currentPower
         if powerDiff<minPower then
         minPower=powerDiff
         allocatedHost=host
         end if
    end if
  end for
  if allocatedHost=null then
      sort the hostList in descending order based on the CPU utilization
      for each host in hostList do
        if host is Suitable for(VM) then
          allocatedHost=host
        end if
      end for
  end if
  allocationMap.add(VM,allocatedHost)
end for
```

Algorithm 2 starts from the initial stage where the initial temperature parameters are set and the initial allocation map of VMs is done by using algorithm 1. Then, many allocation maps are found by performing a lot of cycles. At the end of each cycle, the temperature value is reduced. When the temperature is below the given stop temperature, the algorithm is finished. During each cycle, the algorithm randomly selects two VMs from the current allocation map which is used for reallocation, and a new VM allocation map is generated through Algorithm 1, and the new objective value can be calculated based on the formula (4). Next, the SA method decides whether to switch to the new state or stay in the current state according to the state transition function. If the new objective value is smaller than the current objective value, the new VM allocation is selected to carry out the next round of circulation， otherwise, the new VM allocation will be accepted at a certain probability related to both the temperature and the difference between two objective values. The jump probability of the SA algorithm gradually decreases with the decrease of temperature. The bigger the objective value difference, meaning the worse the new objective value, the smaller the probability. So, the state transition function is calculated as follows:

$$p = \begin{cases} 1, & newObj < currObj \\ e^{\frac{newObj - currObj}{temperature}}, & newObj > currObj \end{cases} \quad (14)$$

It is noticed that a good solution is often obtained during annealing, but it is abandoned due to the high temperature. For this reason, we record the optimal solution of all attempts and take the best solution as the global optimal solution.

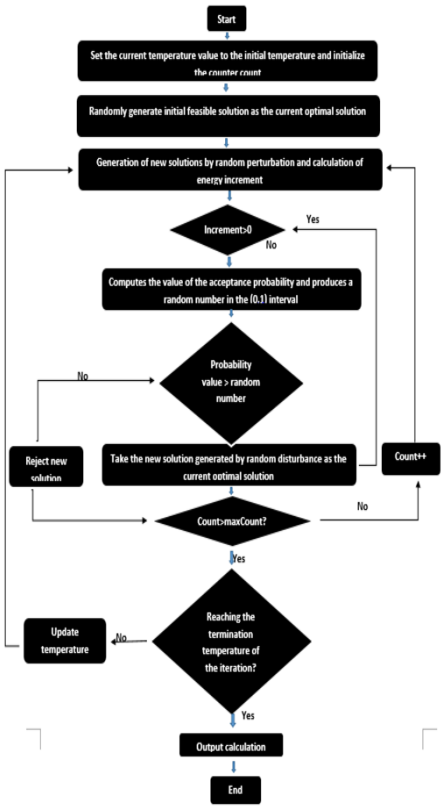

Figure 2: Flowchart of Simulated Annealing Algorithm

Algorithm 2 Simulated Annealing Algorithm

Input: hostList,VMList,initial_temp,stop_temp
Output: allocationMap of VMs
bestAllocationMap=null
bestObjValue=0
curr_temp=initial_temp
Get currAllocationMap based on algorithm 1
currObjValue=0
while curr_temp>= stop_temp do
RandomSelectTwoVMs as the input VMList of algorithm1
Get newAllocationMap based on algorithm 1
calculate newObjValue based on formula (4)
r=random(0,1)
Calculate state transfer probability p based on formula (14)
if p>r then
currAllocationMap=newAllocationMap
currObjValue= newObjValue
end if
if currObjValue<bestObjValue then
bestAllocationMap=currAllocationMap
bestObjValue=currObjValue
end if
curr_temp=0.95* curr_temp
end while
return bestAllocationMap

5. ALGORITHM EVALUATION

5.1. Experimental Setup

We use the CloudSim platform to run our algorithm and verify the effectiveness of the algorithm. Our experiment was performed with a single data center, which consists of 200 heterogeneous hosts, and the configurations are listed in table 1. Considering the heterogeneity of VMs, we have divided the VMs into four types in the experiment, and the experimental setting is shown in Table 2. Suppose that the number of VMs that users need to apply is from 100 to 600, and the CPU utilization of each VM obeys the positive distribution of random variables. In the experiment, we set the initial temperature to 200℃， the stop temperature to 1℃, and the upper threshold value to 0.8, the lower threshold value to 0.2.

Table 1. Experimental values for hosts

CPU(MIPS)	600,700,800,900
RAM(MB)	2000,2500,3000,4000
BW(MBPS)	1000,1400,1600,2000
$Power_{static}$(W)	200,300,400,500
$Power_{dynamic}$(W)	500,600,700,800,900,1000

Table 2. Experimental values for VMs

CPU(MIPS)	50,80,100,120
RAM(MB)	50,90,110,150
BW(MBPS)	100,150,250,300

5.2. Experimental Results

We compare the VM dynamic scheduling method based on workload prediction and simulated annealing (WPSA) proposed in this paper with the Modified First Fit algorithm (MFF) and the simulated annealing algorithm (SA).
The MFF algorithm first uses the AR technique to predict the future workload of the VM and sorts the VM list according to the prediction results, and then migrates the VM to the first conditional host in turn. The SA algorithm is similar to the WPSA algorithm, but it does not use any prediction technology.

a. Comparison of virtual machine migration times

Figure 3 compares the number of virtual machine migrations when using the WPSA, SA and MFF algorithms for dynamic scheduling of virtual machines with different numbers of virtual machines.
From Figure 3, we can clearly see that the number of virtual machine migrations increases with the increase of the number of virtual machines, and the SA algorithm has more virtual machine migration times than other algorithms. When the number of virtual machines is 400, the number of virtual machine migrations of SA algorithm is 6400, while the number of virtual machine migrations of WPSA algorithm is 3800, and the number of virtual machine migrations of MFF algorithm is 4300.
This is because the SA algorithm only migrates virtual machines according to the current load situation, and when the server load changes, it will cause too many virtual machine migrations. When the prediction algorithm is used, because the changes of the future workload are grasped in advance, unnecessary migration is reduced, and because the ARMA model takes the error into account, it has a better prediction effect than the AR model, so the WPSA can be reduced. More migrations.

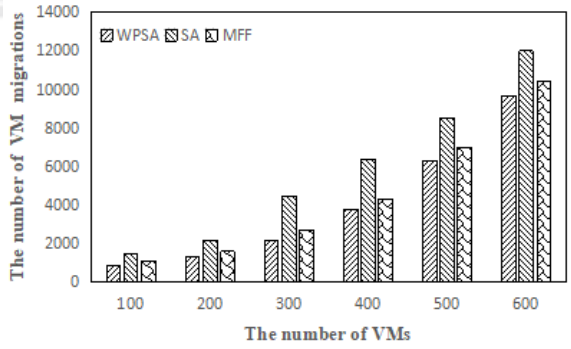

Figure 3: The number of VM migrations with different numbers of VMs

b. Comparison of average SLA violation rates

Figure 4 compares the average SLA violation rate when using WPSA, SA and MFF algorithms for dynamic scheduling of virtual machines with different numbers of virtual machines.

From Figure 4 we can clearly see that the average SLA violation rate of the WPSA algorithm is the lowest.

When the number of virtual machines is 400, the average SLA violation rate of the WPSA algorithm is 8.6%, while the SA algorithm and the MFF algorithm are 10.82% and 9.2%, respectively. This is because the SA algorithm only executes the migration of virtual machines for the servers that are currently overloaded, and does not consider whether the server resources meet the requests of virtual machines in the future, resulting in an increase in the SLA violation rate. Although the MFF algorithm considers the future state of the server, since the prediction effect of the AR model is not as good as that of the ARMA model, the average SLA violation rate of the WPSA algorithm is the smallest.

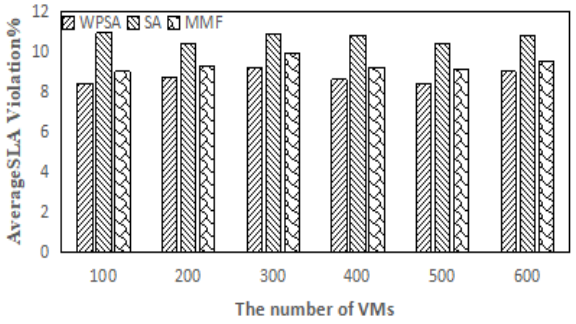

Figure 4: Average SLA Violation with different number of VMs

c. Comparison of data center energy consumption

Figure 5 compares the data center energy consumption when using WPSA, SA and MFF algorithms for dynamic scheduling of virtual machines with different numbers of virtual machines.

From Figure 5, we can clearly see that the energy consumption of the data center increases with the increase of the number of virtual machines, and the energy consumption value of the MFF algorithm is the highest. When the number of virtual machines is 400, the energy consumption of the MFF algorithm The energy consumption of the SA algorithm is 94kwh, and the energy consumption of the WPSA algorithm is 80kwh, which is 14.9% less than the SA algorithm and 34% less than the MFF algorithm. This is because when looking for the destination server, MFF only considers whether the server meets the request conditions of the virtual machine, and does not consider the energy consumption of the data center from a global perspective, resulting in the migration of the virtual machine to a server with high energy consumption. The SA algorithm and the WPSA algorithm consider the energy consumption after virtual machine migration, migrate the virtual machine to a server with lower energy consumption, and because the WPSA algorithm reduces the number of virtual machine migrations, the final energy consumption generated by the WPSA algorithm is higher than that of the WPSA algorithm. consumption is the least.

The results of VM migration and SLA violation rate are shown in Figure 2 and Figure 3, and WPSA has an advantage over the other two algorithms. As depicted in Figures, there is a relationship between the number of VM migrations and SLA violation rate, which is because excessive migration can reduce stability and increase SLA violation. The SA algorithm determines the VM migration based on the current workload which is not suitable when the workload changes. We argue that the migration decision depends on current and future workload. When we use the prediction algorithm, we can grasp the future workload changes ahead of time, and first move the future workload increasing VM out of the overload host, which can reduce the number of migrations and reduce the SLA violation rate. We also see that WPSA has a slight advantage over MFF, which indicates that ARMA has better prediction ability than AR because ARMA takes account of the error.

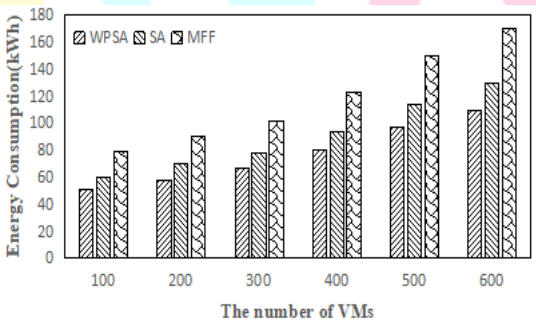

Figure 5: Energy Consumption with different number of VMs

Figure 5 shows the impact of different numbers of VMs on energy consumption. It can be seen that the energy consumption of the three algorithms increases with the increase of the number of virtual machines, and the value of MFF is higher than the other two algorithms. This is expected because when searching for the destination host, the MFF algorithm only considers whether the host is suitable for placement so that the VM is allocated to an inappropriate host. However, the SA algorithm takes into account the energy consumption after the VM is placed, thus preventing more energy waste. Results show that WPSA saved 35% energy compared with MFF and 15% compared with SA.

6. CONCLUSION AND FUTURE WORKS

In a cloud computing environment, resource scheduling strategy directly affects the performance of the whole cloud computing system. Aiming at the problem of energy consumption and considering the heterogeneity of the host, the VM dynamic scheduling method based on workload prediction and simulated annealing (WPSA) is proposed. First, the workload of the VM in the cloud computing system is predicted based on the history load information, and the migration VM is selected according to the prediction results. Then, we use a simulated annealing algorithm to select the target host. Experimental results show that the WPSA algorithm can effectively reduce the number of VMs migration and reduce the total energy consumption of the system. In our future work, we will consider the energy cost of the router in the process of virtual machine migration.

DISCLOSURE

No conflicts of interest in this work.

REFERENCES

[1] Beloglazov A, Buyya R, Lee Y C, et al. A Taxonomy and Survey of Energy-Efficient Data Centers and Cloud Computing Systems [J]. Advances in Computers, 2010, 82:47-111.

[2] Ye K, Jiang X, Huang D, et al. Live Migration of Multiple Virtual Machines with Resource Reservation in Cloud Computing Environments[C]// IEEE International Conference on Cloud Computing. IEEE, 2011:267-274.

[3] Mark C C T, Niyato D, Chen-Khong T. Evolutionary Optimal Virtual Machine Placement and Demand Forecaster for Cloud Computing[C]// IEEE International Conference on Advanced Information NETWORKING and Applications. IEEE Computer Society, 2011:348-355.

[4] Nakada H, Hirofuchi T, Ogawa H, et al. Toward Virtual Machine Packing Optimization Based on Genetic Algorithm[C]// International Work-Conference on Artificial Neural Networks. Springer-Verlag, 2009:651-654.

[5] Kaur T, Chana I. Energy Efficiency Techniques in Cloud Computing: A Survey and Taxonomy [J]. Acm Computing Surveys, 2015, 48(2):1-46.

[6] Cao Z, Dong S. An energy-aware heuristic framework for virtual machine consolidation in Cloud computing [J]. Journal of Supercomputing, 2014, 69(1):429-451.

[7] Arianyan E, Taheri H, Sharifian S. Novel heuristics for consolidation of virtual machines in cloud data centers using multi-criteria resource management solutions [M]. Kluwer Academic Publishers, 2016.

[8] Forsman M, Glad A, Lundberg L, et al. Algorithms for automated live migration of virtual machines [J]. Journal of Systems & Software, 2015, 101(C):110-126.

[9] Li X, Wu J, Tang S, et al. Let's stay together: Towards traffic aware virtual machine placement in data centers[C]// INFOCOM, 2014 Proceedings IEEE. IEEE, 2014:1842-1850.

[10] Beloglazov A, Buyya R. Adaptive threshold-based approach for energy-efficient consolidation of virtual machines in cloud data centers[C]// Proceedings of the 8th International Workshop on Middleware for Grids, Clouds and e-Science. ACM, 2010:1-6.

[11] Dong D, Herbert J. Energy Efficient VM Placement Supported by Data Analytic Service[C]// Ieee/acm International Symposium on Cluster, Cloud and Grid Computing. IEEE, 2013:648-655.

[12] Lu T, Chen M. Simple and effective dynamic provisioning for power-proportional data centers[C]// Information Sciences and Systems. IEEE, 2012:1-6

[13] Talaat M. Short-Term Load Forecasting Using Curve Fitting Prediction Optimized by Genetic Algorithms [J]. International Journal of Energy Engineering, 2012, 7(6):23-28

[14] Li X, Qian Z, Lu S, et al. Energy efficient virtual machine placement algorithm with balanced and improved resource utilization in a data center [J]. Mathematical & Computer Modelling, 2013, 58(5–6):1222-1235.

[15] Marotta A, Avallone S. A Simulated Annealing Based Approach for Power Efficient Virtual Machines Consolidation[C]// IEEE, International Conference on Cloud Computing. IEEE Computer Society, 2015:445-452.

[16] Fan Z, Shen H, Wu Y, et al. Simulated-Annealing Load Balancing for Resource Allocation in Cloud Environments[C]// International Conference on Parallel and Distributed Computing, Applications and Technologies. IEEE, 2014:1-6.