Research of Dynamic Scheduling of Resources Based on Virtual Machine Migration in Cloud Environment

Mo-Han ZHANG\textsuperscript{1,2,a}, Hong-Lin REN\textsuperscript{1,2,b} and Chun-He XIA\textsuperscript{1,2,3,c}

\textsuperscript{1}Beijing Key Laboratory of Network Technology, China
\textsuperscript{2}School of Computer Science and Engineering, Beihang University
\textsuperscript{3}School of Computer Science and Information Technology, Guangxi Normal University

\textsuperscript{a}zhangmohan@buaa.edu.cn, \textsuperscript{b}magia@buaa.edu.cn, \textsuperscript{c}xch@buaa.edu.cn

Keywords: Cloud Computing, Resource Scheduling, Load Balance, Power Consumption

Abstract. Now Cloud data centers exist problems of load imbalance and high energy consumption. The paper studied the resource dynamic scheduling strategy in cloud environment based on the virtual machine live migration and proposed an effective virtual machine target node selection strategy. The algorithm chose the best migration node by calculating the resource demand of the virtual machine for the node to measure the matching degree between the virtual machine and the nodes. Finally, we implemented the scheduling strategy on the cloud simulation platform CloudSim. The experiments results show that the strategy proposed by the paper can improve the load balance of the cluster, reduce the energy consumption of the data center to a certain extent compared with other scheduling algorithms.

Introduction

With the development of economy, service has become the inevitable trend of industrial development, and the results of various production activities has gradually begun to be provided to users in the form of services. Software-as-a-Service (SaaS) [1], Platform-as-a-Service (PaaS), Infrastructure-as-a-Service (IaaS) proposed by the cloud computing are an important manifestation of the trend of service. The cloud computing is developing rapidly in the world by its convenience, economy, easy to monitor and other characteristics.

At present, more and more service providers choose to deploy the applications in the open source cloud platforms. However, these cloud platforms are relatively weak in resource scheduling and can’t dynamically adjust resources in real time according to the nodes’ load conditions. With the data to process and the number of servers of data center continually increasing, it’s common that the servers are faced with the situation of load imbalance, causing resource utilization reduction, increased energy consumption and even reducing the service life of the servers. Therefore, the rational scheduling of resources in the cloud environment is one of the main problems in the field of cloud computing.

Based on the above-mentioned problem, this paper studies the resource scheduling strategy in the cloud environment based on the live migration technology of the virtual machine, and proposes an effective target node selection strategy of virtual machines to enhance the load balance and reduce power consumption during the running of the data center.

Related Work

At present, the research on resource scheduling strategy for cloud environment is mainly focused on two aspects: load balancing and power saving. Zhao et al. in [2] proposed a distributed resource scheduling algorithm COMPARE_AND_BALANCE, which consumes short time and achieves load balance. But it only considered the CPU and I/O resources without the memory, hard disk, bandwidth and other resources. Hermenier et al. in [3] studied how to allocate virtual machines and choose target physical hosts. They proposed an integrated management scheme of virtual machines named
Entropy. This scheme takes into account the impact of VM resource allocation time and migration counts on resource scheduling, which can effectively reduce the numbers of running hosts, reduce the migration counts and migration time of VM and decrease the migration cost.

Beloglazov and Buyya have done a lot of research work to reduce data center energy consumption in [4-6]. They proposed Dynamic Thresholds algorithm in [4] and MBFD (Modification of the Best Fit Decreasing) algorithm in [5] to optimize the scheduling strategy of data center. They consolidate VMs on a small number of hosts using live migration, shutdown the idle physical machines to decrease the number of running hosts and reduce the energy consumption of the data center. Yukinori et al, proposed a scheduling model for VM energy saving in [7]. The model includes four algorithms: prediction algorithm, on/off switch algorithm, task scheduling algorithm and evaluation algorithm. The prediction algorithm predicts the future load demand according to the historical load based on the neural predictor technology. The switch algorithm dynamically switches the computing nodes according to the predicted load data. The scheduling algorithm is responsible for accepting the users’ request and assigning it to the most suitable VM. The evaluation algorithm is responsible for finding a better training cycle and adding or subtracting additional physical hosts as needed.

**Key Definitions**

**Definition 1 WorkLoad:** In this paper, the workload of a node is defined as the weighted sum of CPU, memory, and bandwidth utilization rate[8], the weighting factor W = [w1, w2, w3]. The specific definition is as follows:

\[
\text{WorkLoad} = w1 \times \text{CPU\_usage}(t) + w2 \times \text{Mem\_usage}(t) + w3 \times \text{BW\_usage}(t) \tag{1}
\]

In formula (1), CPU\_usage(t), Mem\_usage(t), BW\_usage(t) denote the utilization rate of CPU, memory, bandwidth and w1+w2+w3=1.

**Definition 2 Power Consumption:** The power consumed when the node is running. Studies have shown that the most of the energy consumed by the node comes from the usage of the CPU and has the following linear relationship with the CPU[8]:

\[
P(t) = kF_{max} + CPU\_usage(t) \times (1-k) \times F_{max} \tag{2}
\]

P(t) is the power consumption of the node at time t, F_{max} is the power consumption at full load, k is the power consumption proportion of full power consumption when the node is idle. From the formula (2), when the F_{max} and k values are constant, the power consumption P (t) depends on the utilization of CPU. The power consumption from idle status to full load status is similar to the linear relationship with the CPU.

**Definition 3 Load Balance:** WorkLoad\_i is the workload of node i, LB represent the load balance of the data center. The LB calculation formula is as follows:

\[
\overline{\text{WorkLoad}} = \frac{\sum_{i=1}^{n} \text{WorkLoad}_i}{n} \tag{3}
\]

\[
\text{LB} = 1 - \sqrt{\frac{1}{n} \sum_{i=1}^{n} (\text{WorkLoad}_i - \overline{\text{WorkLoad}})^2} \tag{4}
\]

n is the number of active hosts, \overline{\text{WorkLoad}} represent the average workload of the data center. LB being larger means that the data center is more load-balancing.

**Scheduler Policy Description**

The dynamic scheduling strategy mainly involves the following three aspects: (1) determination of the time of migration, that is when the nodes begin to migrate; (2) the selection of the VMs to be
migrated, that is which VMs of the source node will be migrated; (3) the choice of the target node, that is which nodes will these VMs be migrated to. Among them, the choice of the target node is particularly important, as it directly affects the load balance level of the data center. If the selected target node is unsuitable, it may lead to invalid migration or secondary migration, increasing the system burden. This paper proposes a BestMatch algorithm for target host selection based on virtual machine matching degree. We use CPU, memory and bandwidth forming a three-dimensional vector \( (CPU, Mem, BW) \) to depict the resources of virtual machines and physical nodes. The resource request vector of \( VM_i \) on the node \( Host_j \) is defined as \( Req_{ij} = (Req_{ijCPU}, Req_{ijMem}, Req_{ijBW}) \), the three components represent the request for CPU, memory and bandwidth. The \( Req_{ij} \) calculation formula is as follows:

\[
\begin{align*}
Req_{ijCPU} &= VCPU_{i,\text{need}} / (CPU_{j,\text{total}} - CPU_{j,\text{use}} - CPU_{j,\text{reserve}}) \\
Req_{ijMem} &= VMem_{i,\text{need}} / (Mem_{j,\text{total}} - Mem_{j,\text{use}} - Mem_{j,\text{reserve}}) \\
Req_{ijBW} &= VBW_{i,\text{need}} / (BW_{j,\text{total}} - BW_{j,\text{use}} - BW_{j,\text{reserve}})
\end{align*}
\]

Where \( i,\text{need} \) is the requested resource of \( VM_i \), \( j,\text{total} \) and \( j,\text{use} \) denote the total resource and used resource of \( Host_j \), \( j,\text{reserve} \) represents the reserved amount of resource, which can be adjusted according to request.

Firstly, the nodes needs to be filtered. Only when the value of \( Req_{ijCPU}, Req_{ijMem} \) and \( Req_{ijBW} \) are all between 0 and 1, the node \( Host_j \) can meet the need of \( VM_i \). If any of the three values is greater than 1, the node can’t be the candidate node, so we remove it from the node set, and the remaining nodes form the resource pool for the VMs to migrate.

Secondly, calculating resource requirements and matching degree of \( VM_i \) for each node \( Host_j \). According to the demand ratio of the CPU, memory and bandwidth of VM, we set the weight vector \( W = (W1, W2, W3) \) and calculate the demand count \( S_{ij} \) of the \( VM_i \) for \( Host_j \):

\[
S_{ij} = Req_{ij} \times W^T, (W1 + W2 + W3 = 1)
\]

In the paper, we get the weight vector \( W \) using the following calculation:

\[
W1 = \frac{VCPU_{i,\text{use}}}{CPU_{i,\text{total}}} \times \frac{VMem_{i,\text{use}}}{Mem_{i,\text{total}}} \times \frac{VBW_{i,\text{use}}}{BW_{i,\text{total}}}
\]

\[
W2 = \frac{VCPU_{i,\text{use}}}{CPU_{i,\text{total}}} \times \frac{VMem_{i,\text{use}}}{Mem_{i,\text{total}}} \times \frac{VBW_{i,\text{use}}}{BW_{i,\text{total}}}
\]

\[
W3 = \frac{VCPU_{i,\text{use}}}{CPU_{i,\text{total}}} \times \frac{VMem_{i,\text{use}}}{Mem_{i,\text{total}}} \times \frac{VBW_{i,\text{use}}}{BW_{i,\text{total}}}
\]

\[
VCPU_{i,\text{use}}, VMem_{i,\text{use}}, \text{and } VBW_{i,\text{use}} \text{ represent the proportion of used CPU, memory and bandwidth of } VM_i \text{ in the source host. Using this method to determine the weight can select a target node with more remaining resources relatively based on the preference of resources for the } VM_i .
\]

Obviously, \( S_{ij} \) is in \((0, 1)\), and the larger \( S_{ij} \) is, the closer the node will be from the upper threshold after the VM is migrated to the node. Such migrations will cause other VMs unable to find a suitable target node. The nodes with more resource can’t receive more virtual machines, which violates the load balancing requirements. Therefore, this paper defines the matching degree as follows:
Similarly, $Match_{ij}$ is in (0, 1) and the larger $Match_{ij}$ is, the higher fitness of the $VM_i$ is for the node$Host_j$, and the better load balance of the system will be achieved.

Now there is still a problem, the node with better performance and more resources can be matched to a relatively large number of migrating VMs, but it’s easy to lead to clustering effect due to that all the migrations happen at the same time, causing the load of the target host getting too high after migration. In order to avoid the occurrence of clustering effect, the paper adopts the method of probability roulette to choose the ultimate node. We define the $p_{ij}$ as the migration probability of $VM_i$ for$Host_j$, and the $p_{ij}$ is determined as follows:

$$p_{ij} = \frac{Match_{ij}}{\sum_{j=1}^{k} Match_{ij}}$$  \hspace{1cm} (13)

In the formula (13), $Match_{ij}$ represent the matching degree of $VM_i$ for$Host_j$, $k$ is the number of the physical nodes after filtered. The $p_{ij}$ satisfies the relations that $\sum_{j=1}^{k} p_{ij} = 1$.

Each VM can get a probability roulette according to the migration probability of different nodes, as shown in Fig.1.

![Selection Probability Roulette of VM](image)

**Figure 1. Selection Probability Roulette of VM.**

Every VM to be migrated corresponds to a selection probability roulette. From Fig.1 we can know that the node with more resource and better performance has larger space in the roulette, so it has larger probability to be selected as the target node. This greatly improves the load balance in the data center and reduce the occurrence of clustering effect to some extent.

**Experiment and Evaluation**

**Simulation Environment Configuration**

In order to verify the performance of the dynamic scheduling algorithm proposed by this paper, we use CloudSim [9] for simulation verification. In our experiment, 100 physical nodes are configured. In order to simulate the heterogeneity of the resources in the cloud platform, the 100 nodes are divided into four groups, each with 25 nodes, and the configurations of the nodes are the same within the group while configurations are different between groups. The specific nodes configuration is shown in Table 1:
Table 1. Experiment Node Configuration.

| Hosts Configuration | Resource Type | Hosts Configuration | Resource Type |
|---------------------|---------------|---------------------|---------------|
|                     | CPU           | Memory              | Bandwidth(MB/s)|
| 1                   | 500MIPS       | 4GB                 | 20            |
| 2                   | 1000MIPS      | 8GB                 | 40            |
| 3                   | 1500MIPS      | 8GB                 | 40            |
| 4                   | 2000MIPS      | 4GB                 | 20            |

We provide 300 virtual machines in total for all nodes, with CPU generated randomly from 100 to 200 MIPS, memory generated randomly from 1G to 2G, the bandwidth generated randomly from 5 to 10Mbps. The 300 virtual machines are initially placed on the 100 physical nodes using the First Fit algorithm. We process 500 independent cloud tasks, with length randomly generated from 30000 to 50000. The simulation runs for more than 20 minutes. We record load balance and total power consumption of the cloud data center every 100 seconds.

Comparison in Load Balance

In formula (1), to calculate the host workload, we set the weight vector $W = [0.5, 0.25, 0.25]$. Monitoring the all nodes’ workload, if the system trigger migration, it begins to migrate VMs dynamically. We compared BestMatch scheduling strategy proposed in this paper with no-Migration strategy, random scheduling strategy and MBFD scheduling strategy, and the load balance of the experiment result is shown in Fig.2.

![Figure 2. Statistics in Load Balance.](image)

Due to different tasks assigned to each node, the workload of each node is different, and the migration count increases with the running of the system. The load balance of traditional no-Migration policy is significantly lower than the other three dynamic migration algorithms. Because MBFD algorithm is only aimed at reducing energy consumption and did not take load balance into consideration, so the load balance is relatively low. The BestMatch strategy proposed in this paper achieves the best load balancing over time. Random scheduling strategy exists strong randomness, and the load balance is also lower than the BestMatch strategy.

Comparison in Power Saving

From formula (2) we know that the energy consumption is mainly related to the CPU utilization rate of the node. In the experiment, we set the energy consumption of the node $F_{max} = 250W$, and the energy consumption ratio $k = 0.7$. The simulation experiments of no-Migration strategy, random scheduling strategy, MBFD scheduling strategy and BestMatch scheduling strategy are carried out respectively. The result is shown in Fig.3.

From Fig.3, we can know that as no-Migration strategy is lack of dynamic resource scheduling, it has little energy optimization. MBFD algorithm is designed for the optimization of energy
consumption, so it achieves the best optimization effect. As BestMatch algorithm uses double threshold migration trigger strategy which shutdown the low workload nodes after migration, it achieves fine energy efficiency optimization effect as well.

![Figure 3. Statistics in Power.](image)

**Conclusion**

The paper studies the scheduling strategy and scheduling algorithm in the cloud environment based on virtual machine migration technology, and solves the problem of the target nodes selection in dynamic scheduling process. Finally, the strategy of this paper is simulated and implemented on the CloudSim. The experiment proves that the scheduling strategy can achieve better effect in load balance and power saving of the cloud data center and has practical significance in the future.

**Acknowledgement**

The project is supported by the Joint Funds of the National Natural Science Foundation of China (Grant No.U1636208) and Aviation Industry Corporation Industry-Academic-Research of China (Grant No.CXY2011BH07).

**References**

[1] Armbrust, Michael, Fox, Armando, Griffith, Rean, Joseph, D. Anthony, Katz, and H. Randy, "Above the Clouds: A Berkeley View of Cloud Computing," Eecs Department University of California Berkeley, vol. 53, pp. 50-58, 2009.

[2] Y. Zhao and W. Huang, "Adaptive Distributed Load Balancing Algorithm Based on Live Migration of Virtual Machines in Cloud," in International Conference on Networked Computing and Advanced Information Management, Ncm 2009, Fifth International Joint Conference on Inc, Ims and Idc: Inc 2009: International Conference on Networked Computing, Ims 2009: International Conference on Advanced Information Management and Service, IDC 2009: International Conference on Digital Content, Multimedia Technology and ITS Applications, Seoul, Korea, August, 2009, pp. 170-175.

[3] F. Hermenier, X. Lorca, J. M. Menaud, G. Muller, and J. Lawall, "Entropy: a consolidation manager for clusters," in VEE 09: ACM Sigplan/sigops International Conference on Virtual Execution Environments, 2009, pp. 41-50.

[4] 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, 2010, pp. 1-6.
[5] A. Beloglazov and R. Buyya, "Energy Efficient Allocation of Virtual Machines in Cloud Data Centers,\" in Ieee/acm International Conference on Cluster, Cloud and Grid Computing, CCGRID 2010, 17-20 May 2010, Melbourne, Victoria, Australia, 2010, pp. 577-578.

[6] A. Beloglazov and R. Buyya, "Energy Efficient Resource Management in Virtualized Cloud Data Centers," in Ieee/acm International Conference on Cluster, Cloud and Grid Computing, CCGRID 2010, 17-20 May 2010, Melbourne, Victoria, Australia, 2010, pp. 826-831.

[7] T. V. T. Duy, Y. Sato and Y. Inoguchi, "A Prediction-Based Green Scheduler for Datacenters in Clouds," IEICE Transactions on Information & Systems, vol. 94, pp. 1731-1741, 2011.

[8] Z. Deng, Z. Duan and L. Li, "Research of Dynamic Scheduling of Resources under the Environment of OpenStack," Journal of Northwestern Polytechnical University, vol. 34, pp. 650-655, 2016 (in Chinese).

[9] 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 & Experience, vol. 41, pp. 23–50, 2011.