

The Processing Strategy of IO-intensive Application in Cloud Environment

Pengfei Zhao¹

Shanghai Institution of Technology, Shanghai, 201400, China

E-mail: zhaopengfei11112@126.com

Lanfeng Zhou

Shanghai Institution of Technology, Shanghai, 201400, China

With the development of cloud computing, how to effectively guarantee the cloud environment with a maximum efficiency and optimize the use of resources and applications has become one of the hot researches. However, application processing in the cloud environment now generally is confined to a common method, i.e., resource allocation, without taking into account different types of applications. In light of this phenomenon, this paper conducts a study to expand the virtual machine scheduling strategy based on a common type of application in the cloud environment, namely the IO intensive application. It puts forward an adaptive virtual machine scheduling policy. The policy can automatically migrate the corresponding number of virtual machines which run IO intensive applications according to the physical server load. Through a lot of experiments, this paper verifies the effectiveness of this strategy and allows the efficiency of processing IO-intensive applications to achieve a certain degree of optimization.

*ISCC 2015
18-19, December, 2015
Guangzhou, China*

¹Speaker

1. Introduction

The emergence of cloud computing has changed the operation mode of the traditional data center, guided the trend of change in IT technology and exerted a tremendous impact on all aspects of the society. Through the technology of virtualization, more and more data centers provide a powerful resource pool for cloud computing. However, how to enhance the efficiency of cloud computing and reduce energy consumption with more efficient use of resources, has become one of the hottest areas in the current research.

Virtualization technology is one of the key technologies which is essential to build cloud infrastructure [1]. At present, for cloud computing data centers, the use of virtualization technology to improve the utilization and efficiency of resources, is primarily focused on two ways: 1) Load balancing: make the user's request in the form of a virtual machine, then allocate to all servers in an average manner so as to make the physical server in the cloud data center to maintain a balanced state which can ensure the processing efficiency of the application[2-4]; 2) Merge virtual machine: make the virtual machine merged and deployed as fullest as possible with time consumption as little as possible on a physical server, and make full use of resources of the physical server, for the purpose of reducing the number of physical servers, thereby reducing the energy consumption of the entire cloud data centers[5].

However, these two ideas are somewhat deficient. As the number of applications continues to increase on a cloud platform and mix different types of virtual machines, it will greatly reduce the processing efficiency. Because physical resources in each physical host are different, and it is possible that one of them will become a resource bottleneck for the processing efficiency. And if one of them does not meet the requirement, the physical host may not accept any one virtual machine deployed, which may result in a great waste in other capabilities of the physical host. So the ultimate purpose of this article research is to develop virtual machines available for different types of cloud applications. (the article studies on IO intensive applications) A strategy is proposed which not only efficiently handles IO intensive applications, but also enables the physical server load in a normal state. The policy given in this paper can automatically migrate the corresponding number of virtual machines which run IO intensive applications according to the physical server load. It is thought that with the virtual machine (the user's request) merger deployed on as few as physical servers and to take advantage of resources of the physical server, it will reduce the number of physical servers, thereby improving the efficiency of applications and reducing the energy consumption to some extent.

2. Related Work

The thought of exploring cloud resource in a cloud environment is mostly based on the mapping between physical servers and virtual machines. The mapping relationship between physical servers and virtual machines (the deployment between physical servers and virtual machines) has been a hot issue in the cloud environment. Research on this issue mostly touches upon two areas. The first is to explore the relationship between physical and virtual machines from the perspective of load balancing; and the second aims at maximizing the utilization of cloud resources, i.e., achieving the optimal use if resources with minimal physical host carrying with as many as virtual machines.

To study the mapping between physical and virtual machines in the cloud environment and in terms of the resource scheduling policy, some scholars have conducted some researches from the perspective of load balancing. As the paper proposed a virtual machine cluster resource scheduling policy, made the virtual machine resources classified and represented by vectors, and described the number of each type of resource and measurement of the load, with the fittest algorithm to find the target host where the virtual machine migrate. Made the cluster of virtual machine achieve the double goal of load balance and energy saving[6]. The another paper proposed a distributed virtual machine migration program aimed at host resource load imbalance in the cloud environment. Through the establishment of the host state domain model, this paper classifies the hosts based on different load conditions, combines the ant colony algorithm, draws the list of virtual machines and the physical host which should migrate, and ultimately makes the cloud platform load in an balanced fashion[7].

In a cloud environment, from the perspective of maximizing the use of cloud resources, many scholars have done some relevant researches. In the paper, Liang-TehLee et al dynamically adjust the CPU utilization of virtual machines on a physical host distribution by monitoring the physical host , thereby reducing the idle load and improving the utilization of CPU resources in the cloud environment[8]. Some other paper through the establishment of utility maximization model of cloud virtual machine resource allocation, provides a higher level of the scheduling choice, and an efficient allocation of quantitative physical resources to place the virtual machine and improve the effectiveness of idle resources[9]. Likewise, some other paper also controls the deployment of virtual machines through setting maximum and minimum threshold value of physical node resources and dynamically merges virtual machines, by stopping idle nodes so as to reduce the number of physical nodes. However, although the above methods can improve the resource utilization, they haven't taken into account the relationship between the application types and physical characteristics of resources between nodes, which cannot solve the problem when only a virtual machine can be deployed on one physical node[10]. While the literature takes into account different types of applications and different tasks of the resource consumption, and conduct live migration of virtual machines, it only involves the cpu and memory resources for migrating the virtual machine[11].

It can be seen that there are abundant researches in this field. However, only few deal with the strategy for IO intensive applications. This paper targets at a variety of resource consumption parameters and characteristics of IO intensive applications in the cloud environment, establishes the virtual machine migration threshold determination model and proposes an adaptive scheduling strategy of IO intensive applications which are deployed on the VMs. So when processing the IO-intensive applications, its handling efficiency can be optimized and the energy consumption can be reduced to some extent.

3.IO-intensive Applications Scheduling Strategy

3.1 Established Model

Firstly, exploring physical limits of the host processing IO intensive applications refers to the fact that how many physical hosts at the same time can be able to suffer IO intensive virtual machines with an uncertain number (how many IO-intensive application tasks) without affecting the rule of SLA service. This paper analyzes resources of physical host which have the following definitions.

Definition 1 :(type of the physical resource) given a physical host, this physical host has a variety of different physical resources, such as CPU resources, memory resources and so on. With a collection R_n (which represents different processing capability) to represent, n represents different resource types of the physical host. For example, R_1 represents CPU resources, R_2 represents memory resources. A collection of R is expressed as:

$$R=(R_1, R_2 \dots R_n \dots R_N)_{n \in (1, 2, \dots, N)} \quad (3.1)$$

Definition 2 :(type of the virtual machine) given a specific application, the cloud data center is assigned to the appropriate virtual machine. The virtual machine will be divided into different types (V_i) depending on the application required resources. Resources required of each virtual machine type (V_i) is different. Resource characteristics $r_{i,n}$ represent the i -type virtual machine's consumption and physical host resources corresponding to the type. i type is designed for IO intensive virtual machines, CPU-intensive and memory-intensive or bandwidth-intensive.

$$V=(V_1, V_2 \dots V_i \dots V_m) \quad i \in (1, 2, \dots, m) \quad (3.2)$$

$$r_{i,n}=(r_{i,1}, r_{i,2} \dots r_{i,n} \dots r_{i,N}) \quad n \in (1, 2 \dots N) \quad (3.3)$$

Definition 3 :(maximum virtual machine limits) because it can deploy multiple virtual machines through virtualization technologies on a single physical host, while without an unlimited number of deployments and limited by the physical host configuration, a single physical host can only carry a certain number of virtual machines k_i at the same time.

$i \in (1, 2, \dots, N)$, k_i represent a number of different types of virtual machines. The most deployed virtual machine condition is expressed by the following formula:

$$\sum_i (k_i * V_i) \leq \alpha_i R_i, 0 < \alpha < 1 \quad (3.4)$$

And by definition 2, it is derived as follows:

$$\sum_{n=1}^N (k_i * r_{i,n}) \leq \alpha_i R_i, 0 < \alpha < 1 \quad (3.5)$$

where α_i is a correlation coefficient. Each physical host for the system, hardware and other factors, utilization of system resources, will not reach 100%, so according to the actual situation, there will be a parameter α_i to identify relevant resources usage limit.

This paper studies the related scheduling policy based on IO intensive applications. As to the limit of resources, it can focus on the most important resource consumption, namely IO resources. Therefore, it establishes the following formula:

$$\sum_{n=1}^N (k_{IO} * r_{IO,n}) \leq \alpha_{IO} R_{IO}, 0 < \alpha_{IO} < 1 \quad (3.6)$$

where $r_{IO,n}$ represents resource characteristics of IO type virtual machines that consume the main parameters of IO resources, n is different parameter that represents different IO resource consumption, k_{IO} indicates the number of virtual machines which run IO intensive applications;

$\sum_{n=1}^N (k_{IO} * r_{IO,n})$ represents IO resources of virtual machines that can be turned on; $\alpha_{IO} R_{IO}$ represents IO resources of a physical host. So the open IO resources of virtual machines cannot exceed the IO resources placed on the physical host.

Derived from the above definitions: when the cloud data center includes a group of IO intensive applications, and as long as IO intensive virtual machines on a physical host conform to the formula (3.4), (3.5), (3.6), the physical host can receive new IO-intensive applications. On the contrary, because of the lack of physical resources, it cannot create a new virtual machine. That is when this physical host virtual machine is turned to the maximum number.

In this paper, according to the change of IO load and IO intensive application processing time, it can set on a single physical host for how much of the virtual machine is running. For the IO intensive application, this paper sets the threshold of IO intensive application in the virtual machine at N , the physical host processing efficiency of IO intensive applications at E_{IO} . k_{IO} represents the number of virtual machines which run IO intensive applications. When $k_{IO} \leq N$, then the E_{IO} is within a reasonable range; and when $k_{IO} > N$, the E_{IO} efficiency drops sharply, and does not accord with the requirement of users. The threshold N is used to assess whether E_{IO} is in a reasonable scope of efficiency. If not, follow the process of IO intensive application migration. But if it is within a reasonable range, it can be divided into the efficiencies of stretching resources range and resources with least use. Here it will maximize the use of resources of the virtual machine when the threshold is set at M ; when $k_{IO} < M$, argues that the physical machine has not made full use of resources, it presents the existence of idle resources.

Therefore, based on the above analysis, the efficiency of physical machines running IO intensive applications with specific use E_{IO} is expressed as follows:

$$E_{IO} = \begin{cases} 1, & M \leq k_{IO} \leq N \\ 0, & k_{IO} < M \\ -1, & k_{IO} > N \end{cases} \quad (3.7)$$

In measuring IO intensive applications processing efficiency E_{IO} , the processing efficiency can be seen directly from the change time of processing IO intensive applications. However, when processing applications stay in a cloud environment, they cannot carry out the scheduling policy by the completion time of running a virtual machine, and this cold migration will only increase the system load, but reduce efficiency. Therefore, in order to find the carrier with IO intensive virtual machines of maximum threshold N , and the optimum processing efficiency threshold corresponding to virtual machines on a physical host M , a comprehensive measure with the following parameters able to withstand the load threshold value of the IO and IO intensive virtual machines on a single physical host is taken.

1: loadavg: namely CPU load. It represents a period of time waiting for the CPU processing and handling of statistical information and the number of processes, which holds the CPU usage statistics of the queue length. The maximum general loadavg in the system is 2. High system load indicates performance problems. Generally the system load is an important performance parameter which reflects the cpu load that is the high system load indicating the lack of the CPU capacity or the CPU number is not enough. However, the high load simply indicates too many running queue. But the task queue may actually be CPU-intensive, or consume IO resources. This paper only aims at IO-intensive applications to research the situation of the physical host in the cloud environment. So the parameter of loadavg can be a standard to measure the IO performance of the physical host.

2: *runq-sz*: that indicates the length of the running queue or the number of processes waiting to run. The maximum value is 2. Both of two parameters measure the situation of the cpu load, but the parameter of *runq-sz* is different from the *loadavg* parameter. It reflects the change situation of the physical host's instantaneous load, while *loadavg* reflects the past 1min, 5min, 15min average load. So the instantaneous reference value has more significance.

3: *avgqu-sz*: refers to the average IO queue length. Queue length (*avgqu-sz*) can also be used as a measure of the system IO load index. When the IO queue length becomes larger, it indicates that system IO bottlenecks exist. Usually the maximum value is 1.

4: *util*: how many percent of the time in one second are for IO operations, that is, the utilization of IO operations in one second, or how much time when the IO queue is not empty in one second. The maximum value is 100%. This parameter can reflect how busy the disk works.

5: *contexts*: namely the context switching. Excessive context switching will cause a lot of system overhead. In the system, this article aims at running IO intensive applications, so it requires the short cpu time slice and needs a lot of context switching. For IO intensive applications, the system of values in the context switch is typically 15,000 to the maximum.

Because an isolated parameter does not accurately reflect the system io load and the physical host performance situation, so comprehensive consideration is given to the above five parameters. Here more than a few parameters can be defined as quintuple threshold decision models. They are defined as following models.

$$T=(loadavg,run,avgqu,util,contexts) \quad (3.8)$$

The specific virtual machine migration threshold model based on IO-intensive applications is shown in Table 1.

Parameters	Maximum threshold	Optimal threshold range	The number of virtual machines with an optimal threshold value range
loadavg	$loadavg_{max_i}$	$loadavg_A \leq loadavg \leq loadavg_B$	$(M_{loadavg}, N_{loadavg})$
run	run_{max_i}	$run_A \leq run \leq run_B$	(M_{run}, N_{run})
avgqu	$avgqu_{max_i}$	$avgqu_A \leq avgqu \leq avgqu_B$	(M_{avgqu}, N_{avgqu})
util	$util_{max_i}$	$util_A \leq util \leq util_B$	(M_{util}, N_{util})
contexts	$contexts_{max_i}$	$contexts_A \leq contexts \leq contexts_B$	$(M_{contexts}, N_{contexts})$

Table 1: Virtual Machine Migration Threshold Model Based on IO-intensive Applications

In terms of the optimal threshold range in Table 1, the model just abstractly gives the optimal threshold range, and in order to determine the specific threshold interval, this article sets a maximum threshold of 50% -70% as the optimal threshold range. Corresponding to each parameter value, both the cpu and io resources, when its cpu or io load exceeds the maximum of 70%, then the efficiency is reduced, and when the load is less than 50% of the maximum load, then the resource utilization rate is low. So we can draw the corresponding optimal threshold value interval of virtual machines through five parameters with an optimal threshold value interval in the model.

However, this paper is to comprehensively consider several parameters to analyze changes of IO resource utilization in the physical host. Therefore, this article, through the intersection of the above five optimal threshold intervals in Table 1, reduces the error of the optimal threshold value of VM between 50% to 70%, and gets the final optimal interval $[M, N]$ (ie this time $E_{IO}=1$). The following equation (3.9) is shown below. The same method can be applied to $E_{IO}=-1$ and $E_{IO}=-1$ corresponding to the virtual machine threshold range.

$$(M, N) = (M_{loadavg}, N_{loadavg}) \cap (M_{run}, N_{run}) \cap (M_{avgqu}, N_{avgqu}) \cap (M_{util}, N_{util}) \cap (M_{context}, N_{context}) \quad (3.9)$$

3.2 Strategy Steps

Through the merger, as many as virtual machines are deployed on as few as physical servers by taking advantage of the physical server resources, so as to reduce the number of physical servers, make the processing efficiency of the application in the physical host reach the optimal, and reduce the entire cloud computing data center's energy consumption. By the upper section of each parameter and model proposed for IO intensive applications, the virtual machine scheduling policy is developed. Specific steps are as follows. The specific flow chart is shown in Figure 1.

Pretreatment: Set Table: S1: the set of physical servers when $E_{IO}=0$; S2: the set of physical server when $E_{IO}=1$; S3: the set of physical servers when $E_{IO}=-1$; S4: the set of the remaining unopened collection of physical servers.

Step1: initialize the list of S1-S4, and wait for a long period of IO-intensive applications. If the application is part of IO intensive applications, perform Step2.

Step2: determine whether S1 is empty, if empty, skip to Step3; if not empty, then send the application to the opening virtual machine in the S1 physical host. Then update S1-S4, do the physical host determination of the threshold model, and do the associated virtual machine migration if necessary.

Step3: determine whether S4 is empty. if empty, then refuse to distribute, waiting to be processed, and skip Step2; if not empty, then start a new physical server, update S1-S4, and continue to receive IO-intensive applications. Determine the load of physical host. Migrate virtual machines when $E_{IO}=-1$.

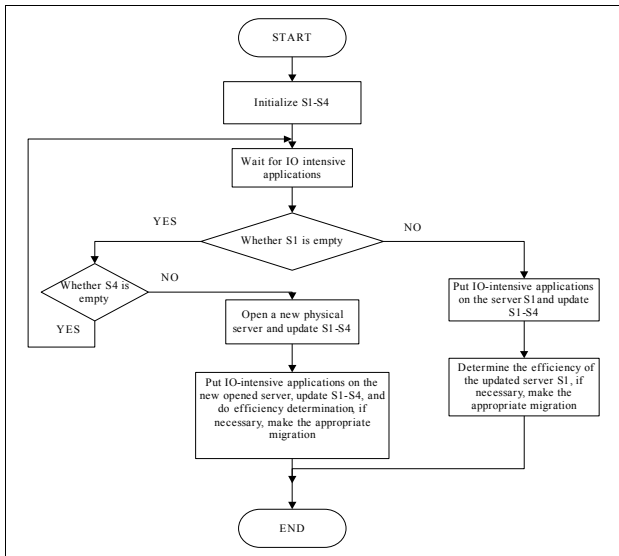


Figure 1: Migration Strategy Process for the Whole Cloud Platform

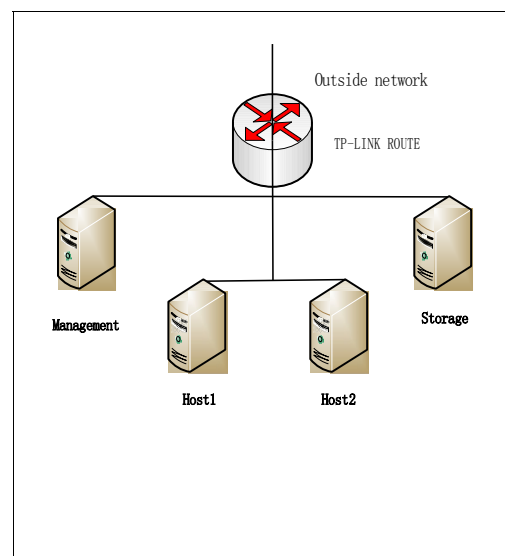


Figure 2: Network Topology

4. Experimental Results and Analysis

4.1 Experimental Environment

In this paper, the experiment is based on the cloud platform named the cloudstack. The whole experiment network architecture diagram is shown in Fig. 2. This experiment cloudtack platform hypervisor uses the KVM virtualization technology; the storage solution on the storage nodes adopts the NFS + Mysql technology.

4.2 Experiment of IO-intensive Application Scheduling Strategy

Aiming at IO intensive applications, the experiment uploads a certain size of files via FTP to simulate the IO intensive application. The same size virtual machine is selected in the experiment (ie CPU to 500MHZ, memory is 512MB, disk to 30GB), in order to monitor the IO load in the physical host and open a sufficient number of virtual machines on the cloudstack platform respectively. Then it simultaneously uploads files whose size is 1GB on each power-of the virtual machine. In this study, the maximum opened virtual machine number is 9. The experiment is repeated for 20 times. Firstly, the paper can indicate the efficiency of opening virtual machines running IO intensive applications through observing the file uploaded completed time. However, as previously mentioned, it is not from the IO-intensive applications processing completion time to determine the physical host IO load. Therefore, this paper determines IO load-related parameters by the Ksar test tool in linux system, then determines how many IO intensive applications are running on the physical host, and whether the efficiency of the physical host processing applications achieves optimal value. Specific experimental results are shown in Fig. 3,4,5,6.

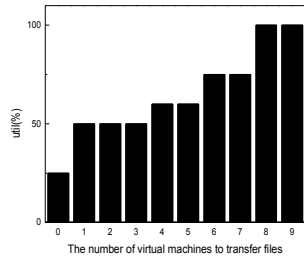


Figure 3: Relationship between the number of VM and disk busy state

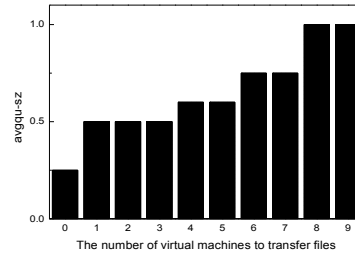


Figure 4: Relationship between the number of VM and IO queue length

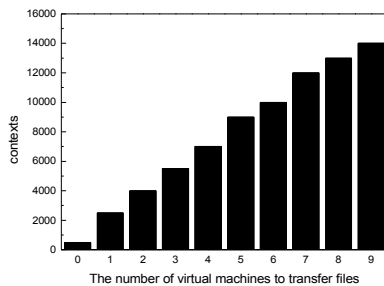


Figure 5: Relationship between the number of VM and context switch

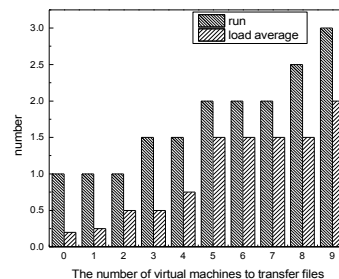


Figure 6: Relationship between the number of VM and run and loadavg

Fig. 3,4,5,6 show that with the increase of the number of virtual machines, physical host performance shows different degrees. In Fig. 3 shows that when the physical host does not accept any IO-intensive applications operation, the physical disk busy extent is 25%, which is in the idle state, and with the physical host running IO intensive applications, the number of virtual machines increases, disk usage continues to increase; when the number of virtual machines stays at 1-3, its disk busy degree doubles; when the number of virtual machines stays at 4-5, the disk busy level continues to increase, indicating that the disk has not fully utilized the IO operation; and when the number of virtual machines stays at 8-9, the disk busy level reaches 100%, indicating that at this time the efficiency of processing IO-intensive applications decreases. It can also be seen from Fig. 5, which is substantially similar to the trend seen in Fig.

4.As can be seen in Fig. 5 and Fig. 6, with the increase of the number of virtual machines, the physical host performance shows different degrees. Therefore, through the determination of the relevant five performance parameters and model T, it can obtain five different threshold ranges of the physical host io load, the specific threshold range is shown in Table 2.

parameter E_{10} VM	util	avqu_sz	contexts	run	loadavg
0	1-5	1-5	1-4	1-3	1-4
1	6-7	6-7	5-6	4-7	5-8
-1	8-9	8-9	7-9	8-9	9

Table 2: Three Virtual Machine Efficiency Threshold Ranges in Each Parameter

It can be seen from Table 2, various parameters corresponding to the interval of the number of virtual machines are under three efficiencies. Through the model T the paper proposeS, it can conduct the determination of the threshold of virtual machine running IO-intensive applications. It can draw the threshold range from Table 3 through a series of intersections as the threshold range is shown in Table 3.

E_{10}	The number of virtual machines through a threshold determination of model T
0	1-3
1	6-7
-1	9

Table 3: Number of Virtual Machine Threshold Table in Three eEfficiencies

Table 3 shows that, under three different efficiencies, it can draw three different virtual machine threshold ranges through the determination of model T. So the paper can conduct the scheduling strategy through this T model.

4.3 Comparison of Strategies

With relevant parameters got by experiment, based on T-model ,we can get the optimal number of virtual machines running IO-Intensive applications on one physical host to ensure that the best efficiency of physical server. To verify the feasibility and efficiency of policy in this article, the following will make a contrast between the virtual machine scheduling strategy the paper proposes and the original schedule strategy under the cloud platform.

In the cloudstack platform, a typical resource scheduling strategy is a firstfit policy. The so-called firstfit policy, that is, to find the physical server node in the list , to meet the requirements of the physical server node assigned to the application firstly;but the firstfit policy does not care about the best treatment efficiency of physical servers ,while just meeting resource constraints of the above definition 3.3 . Hence, the paper makes a contrast between the two policies.

When the number of virtual machines is less than 5, the strategy the paper proposes in this article and firstfit strategy are not much different. So we choose the case when the system processing application number is greater than 5. In order to make the result more clear, under two policies we select the total time and the average time of physical server processing applications to make a comparison, respectively with VM=9 and 10. In therms of the firstfit strategy, all applications (no more than 9) can be dealt with on a single physical server. Due to the processing in a same physical server, the average processing time of application is equal to the total processing time of the physical server. Let T_1 be the total time of physical server

processing applications; and T_2 is the average processing application time., then, $T_1=T_2$. With our strategy, it may need more physical servers (herein represent the number of physical hosts), and the migration time it takes should be credited to the total of physical server processing applications and the average time to process the application. Let t_1 be the total time of the physical server processing applications; t_2 be the average processing time of applications on each physical machine; t_3 be the migration time. Contents of the experiment are consistent with the above experiment. Fig. 8 and Fig. 9 show the result of the experiment.

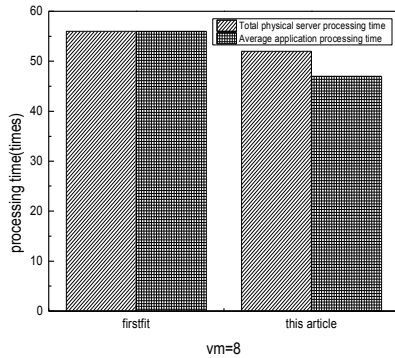


Figure 8: Time comparison of two different strategies when VM=8

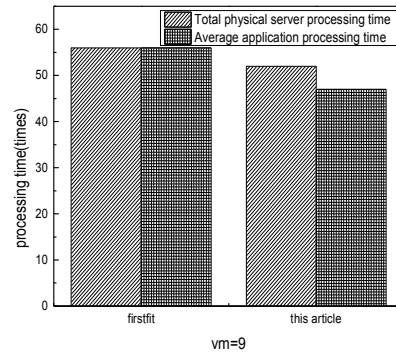


Figure 9: Time comparison of two different strategies when VM=9

Fig. 8 -9 show : with the firstfit strategy, T_1 grows to about 52min and 56min, and compared to 48min-- the minimum time we have got in the previous experiment, the processing time increases by 4-6 minutes, and efficiency decreases by 15% . It is processed in a single physical server, so there is no migration time, meaning $T_1=T_2$. With the strategy the paper proposes, t_3 should be credited to the t_1 and t_2 , there is: $t_1=n*t_2+t_3$. We can see from the above four figures, by adopting this strategy, regardless of the virtual machine migration consuming time, the total physical servers processing time is reduced. Contrast to the firstfit strategy, as a whole, the strategy we propose can effectively reduce the processing time for IO-Intensive applications, and improve the processing efficiency. But only judging from the t_2 in this strategy, $t_2=48$ min. Although there is a certain degree of migration, the migration time is almost negligible for IO-intensive applications which the processing time spent is above 10 minutes. Compared to the FirstFit strategy, our strategy spends t_2 little. The key is to be able to make the processing efficiency of the application remain in the optimum range.

It can be concluded that, for a large number of IO-intensive applications which spend a long time, the IO processing efficiency of physical server must be taken into account. If it can process a large number of IO-intensive applications according to the firstfit policy, it will increase the IO load of the physical server, making IO resources of physical servers accelerate aging and its processing efficiency become very low. As a result, it shows the application process is too long, and it is unacceptable for the customer. But when adopting the strategy the paper proposes, while keeping the total processing time of the case, although the number of physical servers increases, and the time of the migration brings unnecessary overhead, the physical IO load of each server is balanced, the average processing time of applications is

within an acceptable scope of users, more importantly, the application processing efficiency of the physical server can maintain a better state.

5. Conclusion

This article discusses processing characteristics of IO-intensive applications in the cloud environment and presents an adaptive scheduling policy. The main purpose of this policy is to control the number of IO-intensive applications in the physical server by merging or migrating the virtual machine, so as to achieve the purpose of controlling the energy consumption and improve the processing efficiency of IO-intensive applications. Meanwhile, through a lot of experiments in the real environment, it shows that compared to traditional methods, the processing efficiency of IO-intensive applications has remained the best and the IO load of physical server has maintained in the normal range, avoiding the low efficiency of processing IO-intensive applications and the IO overload of physical server when IO-intensive applications increase.

References

- [1] M. Ullrich, Lassig et al. Current Challenges and Approaches for Resource Demand Estimation in the Cloud. International Conference on Cloud Computing and Big Data[J]. 2013, 5(4):387-394.
- [2] H. Wang, Zj. Wang, Ll. Xie. Software virtual methods and application base on Internet [J] System simulation technology, 2014, 6 (8): 56-58(in Chinese)
- [3] R. Soares Boaventura, K. Yamanaka, G. Prado Oliveira. Performance Analysis of Algorithms for Virtualized Environments on Cloud Computing [J]. Latin America Transactions, IEEE (Revista IEEE America Latina), 2014,12(4):792-797.
- [4] D. Loreti, A. Ciampolini. A distributed self-balancing policy for virtual machine management in cloud datacenters[C]. High Performance Computing & Simulation (HPCS), 2014 International Conference on. IEEE (Revista IEEE America Latina), 2014:391 - 398.
- [5] M.F. Li, J.P. Bi, Z.C Li. Resource scheduling overhead waiting perceived virtual machine consolidation [J]. Journal of Software,2014,7(35):1388-1402.(in Chinese)
- [6] W.Y. Zhou, H.P. Chen, etl. Based on the virtual machine migrated virtual machine cluster resource scheduling [J]. Huazhong University of Science and Technology (Natural Science), 2011, 39(1): 130-133(in Chinese)
- [7] B. Guang, Wang Ma. Tang, S. Lei. Study load balancing for distributed virtual machine migration cloud environment [J]. Computer Applications and Software, 2013, 30(10): 87-91(in Chinese)
- [8] L.T. Lee, K.Y. Liu, H.Y. Huang, C.Y. Tseng. A Dynamic Resource Management with Energy Saving Mechanism for Supporting Cloud Computing [J]. International Journal of Grid and Distributed Computing. 2013,6(1):67-76.(in Chinese)
- [9] X. L. Shi. Utility maximization model of cloud virtual machine resource allocation [J]. Journal of Computers, 2013, 36(2): 252-261(in Chinese)
- [10] M. Khushbu, S. Richa, Energy Conscious Dynamic Provisioning of Virtual Machines using Adaptive Migration Thresholds in Cloud Data Center[J]. International Journal of Computer Science and Mobile Computing. 2013,2(3):74-82.
- [11] I. Moreno, S. Moreno, R. Yang, J. Xu, et al. Improved energy-efficiency in cloud datacenters with interference-aware virtual machine placement[C]. Autonomous Decentralized Systems (ISADS), 2013 IEEE Eleventh International Symposium on. IEEE (Revista IEEE America Latina), 2013:1-8.