

Research and Implementation of Load Balancing Technology for Cloud Computing

Sun Hong¹, Wang Weifeng², Chen Shiping³ and Xu Liping⁴

**¹University of Shanghai For Science and Technology, China,
sunhong@usst.edu.cn**

**²University of Shanghai For Science and Technology, China,
wwfhuo@163.com**

**³University of Shanghai For Science and Technology, China,
chensp@usst.edu.cn**

**⁴University of Shanghai For Science and Technology, China,
Email:5850487@qq.com**

Abstract. This article selects load balancing system technology to analyze, combines the live migration technology of virtual machine, and studies the frame of virtual machine live migration as well as the mathematical model applied to concrete process of the migration. The article presents that the process of combining the specific strategies of load balance to the frame of live migration, that the simulation experiment and conclusion to this total process. The study takes Eucalyptus as experimental platform, and decides initial to take Xen as experimental virtual machine. The article's innovations are to optimize the design of virtual machine live migration in cloud environment, and to combine the specific strategies of load balance to the process of live migration. After designing and analyzing the specific strategies of load balance modularization, according to rationalizations that the technology of live migration can apply to the specific strategies of load balance, the study sets the measure index to evaluate load balance, and solve consequently the problem of load in cloud environment, which is realized in the process of live migration. It turned out that algorithm fusion raised has obvious performance advantages.

Keywords: Cloud Computing, Visualization, Modularization, Live Migration, Load Balance

1. Introduction

With the fast development of the Internet and network, people rely more and more increasingly on the access to the Internet for getting information. Amount of data transmission in network and user request appears exploded increase, which raises a higher claim to server processing ability, and makes server sent answer-response in the shortest possible time on the basis that servers accept reasonably client request to improve user experience optimization. A huge amount of data and access request does cry for updating servers, which can response fleetly, are easy-to-use, and have high expansibility to expand network bandwidth in response to user request timely. The appearance of virtual machine live migration is the very effective approach to settle load imbalance. By means that the total virtual machine running state can mutual transfer smoothly between two physical hosts of random clusters, of course, which is processing and does not have any stagnant feelings for the users when it is necessary to migrate, virtual machine live migration can help cloud environment maintainer make full use of node server in the

cluster and dynamically achieve load balancing of cloud resources.

The traditional load balance algorithm is based on task control allocation, when applied to the cloud environment, it has some disadvantages: small task control allocation particles, node load conditions vary greatly, load information cannot be updated in a timely manner, load balance algorithm^[1] will be misguided. Task control allocation needs a central dispatcher, which is in full charge of dispatch and migration, the huge amount of dispatch and migration will make central dispatcher busy and cause disorders easily, which is also a major bottleneck of system performance. There will be more computational tasks in cloud computing environment, the complexity of allocation algorithm will face greater challenges and the availability of algorithm needs more attention.

2. Framework Design of Dynamic of Virtual Machine in Cloud Environment

2.1 Basic framework for dynamic migration of virtual machine

The basic migration structure is implemented by four modules including monitor migration, operation migration, freezing and target domain arousal with the four functions of the corresponding modules to achieve^[2]. As shown in figure 1.

Monitor migration module: The primary function of the primary module is to determine the source machine of the migration, the start time of the migration, and the target machine of the migration. The working mode of monitor migration module is determined by the purpose of migration. In order to ensure that the load of each node is balanced, the monitor signal is set in the virtual machine management program, the monitoring load operation of various nodes determines whether the monitor signal needs to migrate.

Operation migration module: The module is the most important module, and undertakes the most migration of the virtual machine and most work of the migration process. After the start of the migration, the module collects running information of the source machine, and at the same time sends “frozen” signal to freezing module to make the source machine downtime. Then the module continues to copy the remaining pages, after the end of the copy, it sends a wake-up call to wake module of the destination machine. this process is the key part of the whole migration process, and directly influences the downtime in the migration process and migration time.

Freezing module: The module is mainly responsible for how to solve the system to provide users with uninterrupted service^[3] to make users feel the service without interruption.

Target domain arousal module: The function of this module is to determine the time to wake up the destination machine, to make sure that the arousal target machine is in line with the source machine on service, and how to maintain the consistency of the target domain and the original domain on service. After downtime, the module is operated to copy the remaining memory page and sends a wake-up call to wake module of the destination machine after the end of the copy.

The direct consequence of downtime is to make the connecting device interrupt, peripheral device cannot connect to the virtual machine, which will certainly cause peripheral service is not timely or presents a variety of transmission errors.

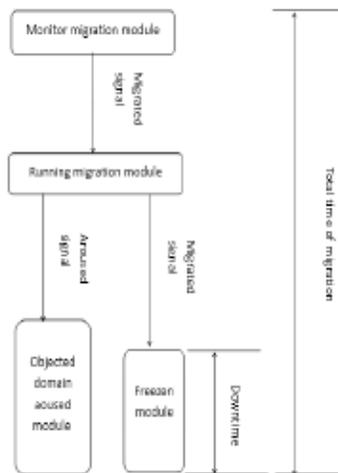


Figure.1 Basic dynamic migration module

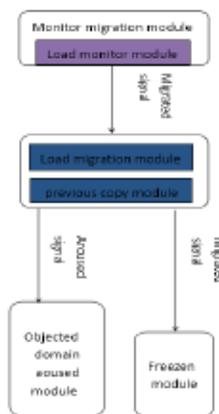


Figure.2 Dynamic migration framework optimization modules

2.2 Dynamic migration framework optimization of virtual machine

In order to increase the rate of load application and make migration process more smooth and effective, we make the frame design of the virtual machine dynamic migration optimized, and add 2 module about the implementation of the load balancing. One load monitor module adds to the original monitor migration module, which is devoted to marking the operation information of current virtual machine, setting the trigger condition for migration, and paving the way for a subsequent migration to select the appropriate load. The other is mainly in charge of location and selection strategy after the start of the virtual machine migration. As shown in figure 2, Grey patterns mark three modules.

3. Implementation of Resource Load Balancing Algorithm in Cloud Computing Environment

3.1 Overview of load balancing

Mutual use of servers is increasingly frequent, because there is a huge gap between the effective use of resources on the servers. Some servers are often in a competition for resources state, some for a long time are rarely even without the use of resources, which greatly reduces the resource utilization of corresponding range of system and leads to the severe decrease of performance of the overall cluster system. Load balance can be directly implemented by hardware level, for example increasing physical

device, and by software level, such as setting the relevant protocol and using specific software. Hardware implementation is to install a load balancer connected to the external network, through which the server for user access in the load application and user can access the resources. Hardware can make the processing capability of the cluster system stronger, at the same time, and promote the equalization performance, but it cannot effectively master real-time status of server, because it parses the data flow only from the network layer, which is not flexible enough.

An effective load balancing algorithm not only be able to assign the load evenly to every server, which can reduce the users' waiting time, but also need to migrate the load above the node over the load value to the node that does not cross the threshold or relatively easy to operate^[4].

Load balancing problem is a classic combinatorial optimization problem. That distribution and redistribution of tasks and resources on each node ultimately make each node load benefit roughly in balance and improve the overall system performance is load balancing. Load balancing algorithm, with respect to the load sharing algorithm, has a higher goal performing the more efficient allocation use of resources.

3.2 Classification of load balancing

Load balancing varies and should be classified on application scope. Classified on the task distribution and redistribution of the nodes to achieve the strategy of the algorithm, the load balancing technology can be classified as follows:

1) Implementation methods of hardware and software

Hardware load balancing primarily implements large system method in the high flow loss, and must increase the specific load balancer, of course, which will have better performance based on the increased cost. However software load balancing is applicable for some small and medium sized web sites or systems, software methods can be very convenient to be installed to the node server. The use of more commonly used URL redirection in computer networks or a technique based on the Internet such LVS to achieve a certain balanced load function can achieve the general equilibrium load demand.

2) Global and local methods

Classified according to the geographical distribution of the server, global load balancing technology is a certain degree of load balancing for multiple servers, which is distributed in the different regions. For access users, global load balancing technology automatically adjusts to the nearest point of the region by determining the location of the IP address. Local load balancing technology can control scheduling process some node of node server cluster in a regional scale at the time to a certain degree and make the node load relatively balanced. The technology can strengthen practical effect through node server designed and make network bandwidth be even distributed to every node server in server cluster.

3) Dynamic and static methods

The load balancing strategy in cloud computing is divided into two types, static state and dynamic state. The so-called dynamic decides from which overload node server chooses task and locates the target node to assign tasks according to the current state of the system. Once a node task of the system is overloaded, some tasks on this overloaded node will be migrated to other nodes to process, and to achieve the results of dynamic equilibrium. Certainly, task migration also brings additional consume

to the system. According to the simple system information, the mathematical function scheduling algorithm is used to select the source node and then locate the destination node and assign tasks and execute, which is the static performance. Static strategy implementation is relatively simple, but it is not fast enough and cannot dynamically adjust the information of each node as far as real-time response, and consequently some nodes utilization is very low. Most of the typical static load balancing strategies are based on the prediction model, for example algorithm based on inheritance, which can predict the trend of nodes according to the current information and historical information of nodes, and then give priority to high available future nodes to resource scheduling and task allocation.

3.3 Cloud resource load balancing strategy

There are a lot of physical servers in the cloud environment and these server specifications are not fully consistent. Through virtualization technology, a single physical node can be modeled as a number of computing machine entities, these virtual machines assign dynamically automatically to the user according to the user's requirements specification. But because of the user's requirement specification is not consistent, and the configuration of all the physical servers in the cluster is also not consistent. Traditional algorithm can balance load to a certain degree, but each algorithm itself have their obvious characteristics and deficiencies and exist such disadvantages, which makes the equalization effect disadvantaged and influences the service performance or causes other problems related^[5]. At present, the main base of load balancing strategy has been divided into 4 types. NO.1 is ratio strategy, NO.2 is minimum number of connections strategy, NO.3 is round-robin strategy, NO.4 is fastest response time strategy.

Ratio strategy installs firstly the external request, and then pre-allocated to each load in a balanced state of the server. Cloud environment system has 4 servers, we can assume that the proportion of the probability of receiving a virtual machine migration request is 2:2:3:1, so each node server processing request is also different. The method is suitable to be responsible that the node server of load balance allocated according to the level of the hardware configuration, and set corresponding ratio depending on the corresponding processing efficiency, so that servers which have not the same properties can also be operated smoothly to prevent some servers overload and others in idle state.

Minimum number of connections strategy is that the hardware device responsible for the balanced effect monitors continuously and checks the number of connections on the relevant node server, selects a minimum number detected node as the purpose node of processing the request, which is suitable for application to long connections.

Round-robin strategy: The scheduler is applied to this strategy regardless of load status of destination node. As long as there is an external request, the request will be distributed to each node server, for example, there are 3 servers in the cluster system, and the request 3 servers will process are the same. Because of handling all transactions on average, this way is only suitable for the same hardware configuration nodes. This mode achieves relatively simple, the algorithm is also very simple to design, and the system overheads less, only to deal with the small business which have small differences and spend a short time.

Fastest response time strategy: The hardware devices send requests to be processed to each node continuously, which node is the fastest on response speed, the request is forwarded to the node server. The algorithm is suitable for stringent real-time response, but this method does not consider the load state of the destination node, which is easily prone to heavy load and causes stress to the

high configure servers.

3.4 Optimization algorithm of resource load balancing

Network nodes often contain memory resources, network bandwidth resources and other key resources. In order to describe accurately the use of various types of resources in the nodes, the load index vector is often used to describe. Each of these components is corresponding to a key resource in the node, which is used to represent the load usage. The algorithms mainly focus on three kinds of resources: CPU, memory and network bandwidth.

To ensure the efficient utilization of resources, we can comprehensively consider the cloud computing system, calculate the load of each server node, combine frame design with dynamic migration, make use of virtual machine migration technology, consider how to select the virtual machine from the overload node how to determine the migration destination and coordinate the load of different servers. This paper focuses on the load balancing part of the module diagram, figure 3 shown in figure.

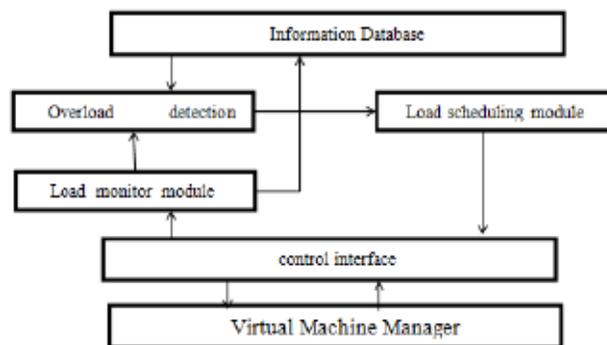


Figure.3 Load balancing algorithm module

Load monitor module: The resources between nodes are heterogeneous. In order to make the data value of the same kind of load can be compared, each component needs to be standardized and description undignified. The CPU load benefit is achieved by using the average utilization rate of all CPU on the nodes as well as the resource usage of the specification of the memory and network bandwidth. The most important of the three load benefits to do the following definition:

1) CPU Load Interest (CLI)

Calculate the average utilization rate of all CPU on the nodes, and reflect the CPU load of nodes by the average utilization rate. The physical CPU number of the node is m , the utilization rate of each CPU is c_i , the CPU load interest CLI of the node is expressed as:

$$CLI = \frac{\sum_{i=1}^m c_i}{m} \quad (1)$$

2) Memory Load Interest (MLI)

Load of a single virtual machine memory includes memory currently in use and memory for paging. The number of virtual machines on the node is m , $Vused_k$ represents the size of the virtual machine memory being used, $VChanged_k$ represents the size of memory of user page, $TotalV$ represents total memory of nodes and is also the sum of the former. Memory load of node:

$$MLI = \frac{\sum_{k=1}^m VUsed_k + VClanged_k}{TotalV} \quad (2)$$

3) Network bandwidth Load Interest (BLI)

The bandwidth load of a node is defined as the ratio of the sum of the bandwidth each virtual machine uses and the total bandwidth. The number of virtual machines on the node is m , $VnetBand_k$ represents bandwidth the virtual machine I is using, $TotalBand$ represents the maximum bandwidth of a node BLI:

$$BLI = \frac{\sum_{i=1}^m VnetBand_i}{TotalBand} \quad (3)$$

Load migration module: The migration process of virtual machine embraces the migration of the original host state and resources(internal memory, CPU and I/O device).Load migration module mainly adopts operation strategies to describe the current load conditions of each server node, and makes sure to record in real time and provides load index used by the server node resources. About the heterogeneous nodes that appear in the system, load operation strategy would also like to unify the description of its standardization and eliminate the heterogeneity, in order to facilitate the use of resources. Load operation strategies have selection rule, distribution rule and location rule^[6].

Usually, virtual machine nodes that need to be migrated often have more than one node in cloud computing systems. Equally, the destination machine of virtual machine migration has more than one server to meet the conditions of acceptance, If we take a physical node with the best performance in the current environment as the most suitable destination machine. All of the overload nodes choose the same destination node to migrate. This is bound to result in a sharp increase in the node load in a short period of time, severe cases will cause the destination node to collapse. The above phenomenon is called cluster effect^[7].

Firstly we identify a set of nodes that meet the low-load, select the destination node from the cluster when locating, and focus on computing node CPU computing and memory capacity of two performance indicators. Besides, if the node appears to have insufficient memory but the CPU computing capability remains or the two are in the opposite, the virtual machine on the physical node can't run properly, and also cause waste of resources. To avoid wasting as much as possible, balance the ratio of memory resources in physical node to CPU computing resources, and achieve optimal utilization, when selecting a destination node, we should mainly consider the matching degree between the virtual machine to be migrated and the destination node. That is the proportion of CPU consumption / memory consumption.

Table 1 Algorithm symbol and meaning

Symbol	Meaning
$(Cmigri) cost$	CPU usage rate of the node N_i in the virtual machine
$(Mmigri) cost$	The memory utilization of the node N_i in the virtual machine
$(C_i) cost$	CPU utilization ratio consumed by N_i
$(M_i) cost$	Memory utilization occupied by N_i
$(C_i) max$	CPU utilization rate migration trigger threshold of Node N_i
$(C_i) available$	CPU available rate of Node N_i
$(M_i) available$	Memory available of Node N_i
$(UCR_i)matched$	UCR matching threshold of node N_i
$Nchosen$	Destination node

$$(R_i)_{available} = Y C_i Y_{available} \times (M_i)_{available} \quad (4)$$

$$(UCR_i)_{available} = Y C_i Y_{available} \div (M_i)_{available} \quad (5)$$

Firstly, according to measurement index $UCR_{available}$ of the target node server, measurement index UCR_{cost} of the virtual machine and the performance of the target node, we can select k destination nodes meeting the requirements from the cluster center. Then the probability model of the $R_{available}$ value of the K labeled nodes is located. Suppose the current available resource capability for node I is $\&$, the probability of the node to accept the migrated virtual machine is P_i . Suppose the current available resource capability for node I is $(R_i)_{available}$. Then through the above description can be learned that the node allows to select the virtual machine to migrate to the probability of P_i as:

$$P_i = \frac{Y R_i Y_{available}}{\sum_{i=1}^k Y R_i Y_{available}} \quad (6)$$

Suppose the destination nodes' set is $\{N1, N2, N3, N4, N5\}$, its capable of utilizing resources $R_{available} = \{2.0, 2.0, 2.0, 3.0, 1.0\}$, and on the basis of the above formula to get the location probability $P = \{20\%, 20\%, 20\%, 30\%, 10\%\}$.

Lastly, when host select a destination node for virtual machine dynamic migration, using a RD random function to generate a arbitrary number in $[0, 1]$. Then according to the number of probability space is in which target node, and make sure the node that virtual machine migration finally choose. The probability that those physical hosts with a strong ability to use resources is being the target node of the virtual machine migrated is also large when there is a virtual machine triggered migration. The lower the utilization ratio of node resources, the greater the possibility of selection as a destination node and the smaller on the contrary. From these examples, we can see that the more idle state a node has and the greater the probability space occupying is, and the chance that generated random number is included in this probability range is larger. Thus the node is more able to be selected as a virtual machine migration destination node. In summary, to a certain extent, the probability mechanism prevents the occurrence of clustering effect, and the load balance of the server cluster in the cloud environment is better realized.

4. Experimental Results Analysis

4.1 Experimental environment and platform

We choose high modularized and rich-interface Eucalyptus suitable for C language as an experimental platform for the cloud environment. Eucalyptus platform is an open source project^[8]. It is a research result to study the global hot topic, cloud computing subject, and put into practice. It implements the IaaS service and enables users to allocate and manage physical resources through Xen or KVM virtualization technology. Eucalyptus interface can be connected to the SOAP and REST interface, if it is a cloud environment based on Eucalyptus platform, other visitors to the non cloud environment uses the SOAP interface or REST interface and can be connected to the common operation. Experiments using 4 PC machines to build a system that can be used as experiment, its topology structure as shown in Figure 4. Detailed configuration of each node as table 2 and table 3.

Table 2 Node hardware configuration

Type	Device	CPU	Memory	Hard disk
PC1	CC Cluster Controller	2.93GHz 32b dicaryon	1.85 G	320G
PC2	NC Node Controller	2.93GHz 32b dicaryon	1.85 G	320G
PC3	NC Node Controller	2.93GHz 32b dicaryon	1.85 G	320G
PC4	NC Node Controller	2.93GHz 32b dicaryon	1.85 G	320Ge

Table 3 Node software configuration

Type	OS	Platform and Module
PC1	Ubuntu10.10	MySQL 5.1+Eucalyptus 3.1+ Load balancing module
PC2	Ubuntu10.10	Eucalyptus 3.1+ Load monitoring module + Anomaly detection module +3th Xen3.4.2
PC3	Ubuntu10.10	Eucalyptus 3.1+ Load monitoring module + Anomaly detection module +4th Xen3.4.2
PC4	Ubuntu10.10	Eucalyptus 3.1+ Load monitoring module + Anomaly detection module +6th Xen3.4.2

4.2 Experimental results and analysis

Summarize the characteristics of the algorithm before the experiment. Firstly, the load balancing module is integrated into the dynamic migration framework, and the previous dynamic migration framework is only limited to the whole process of the migration, which is considered separately from resource scheduling. In this way, when dealing with the migration problem in the cloud environment it is easy to cause a phenomenon that after solving the problem of a unilateral and then producing a new problem; Secondly, new trigger rules for load prediction mechanism are used, which is different from the traditional trigger rules based on specific thresholds, and mainly solves that the transient load peak will trigger the frequent migration. Lastly, we should comprehensively consider the utilization of CPU and memory when selecting rules and locating rules. Especially when positioning rules, this algorithm chooses the probability mechanism used by migration destination nodes and prevents the occurrence of group effect. At the same time it realizes well the load balance of the server cluster in the cloud environment. In addition, the calculation of the location probability of each node does not impact relatively and mutually independent, Under these circumstances the balance of the cloud data center will be better.

The first set of simulation experiment designed carried out the test about migration trigger time. Select one of the nodes to monitor the CPU load utilization rate, and set the threshold of 0.7. Shown as figure 5, when t is equal to 8, once CPU load of nodes exceeds the threshold, the traditional trigger rules will immediately trigger a migration [9]. However, using predict trigger rules can partly predict that the node load is on the decline, which does not trigger the migration. In addition, when t is equal to 40, because the forecast data predicts the next trigger will have the trigger peak, so the first trigger did not immediately exercise. But when t is equal to 43, the trigger is migrated, and compared with the traditional algorithm based on threshold value, three migrations will be triggered during this period and cost vast virtual machine migration resources, which avoids the additional waste.

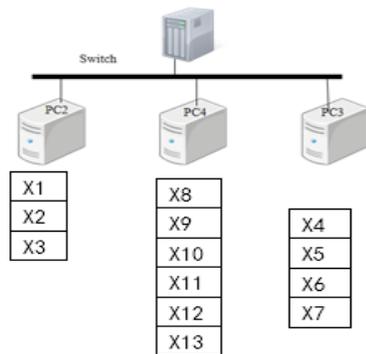


Figure.4 Experimental environment topology

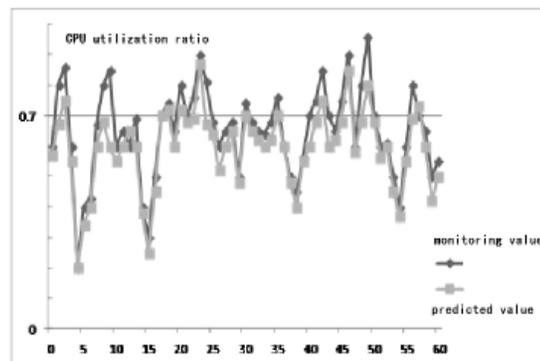


Figure.5 Compared with the traditional methods

In the emulation cloud environment of experiment 2, three machines have the same configuration on the nodes, and have initialized different timely load conditions. we set the use ratio of the system resource 70% and take it as overload limit of the load, at the same time, we stipulate that the request array have the same service type. Initialized load condition of each node need the artificial set, and are set the request array distributed by task scheduling. Each of the calculation of the system resource utilization and the rules of dynamic migration adopts the methods introduced by the above section. The system resource utilization shown as figure 6. We test on selecting a location algorithm of migration destination nodes and take no load balancing algorithm as the baseline, compared with the load balancing algorithm based on optimal adaptive rule. The load balancing of the system is evaluated by the standard deviation of the load benefit. Experimental results are shown in figure 7, figure 8 and figure 9.

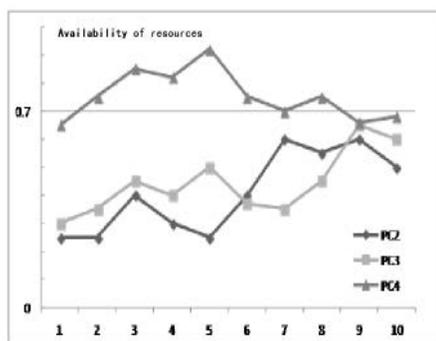


Figure.6 Node load variation

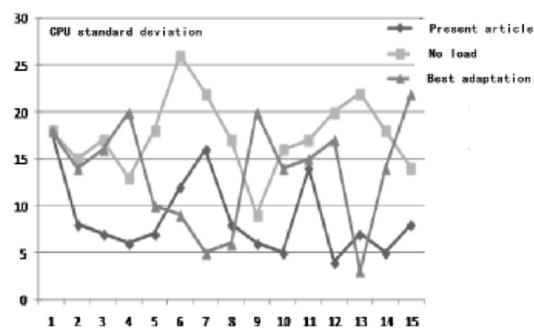


Figure.7 CPU load balancing

In the figure 6, from the initial PC4 start node has been in a super load state, start from the initial PC4 nodes has been in a over-load state, but there is no immediate migration equilibrium which is due to the role of the trigger rule. This experiment also verifies the effectiveness of the trigger rules based on the prediction from the side. When t is equal to 5, PC4 will trigger migration, and migrate the virtual machine to PC2, then the utilization rate of PC2 will obviously increase. Although PC4 has eased, due to the migration of virtual machines, and the utilization rate of PC2 and PC4 was higher than that of PC3. When t is equal to 8. The virtual machine on PC2 and PC4 migrates each load of them to PC3, At the end of this balanced period, the resource utilization rate of each node is similar and in the balanced state. Then approximate load balancing is achieved.

According to figure 7, we can see that CPU load standard deviation distribution is more balanced generally in this algorithm^[10]. Generally, the standard deviation is mainly distributed in the following 10. However the memory and network bandwidth have obvious advantages, the resource of each node in

the colony can be use reasonably. Shown as the follow figure.

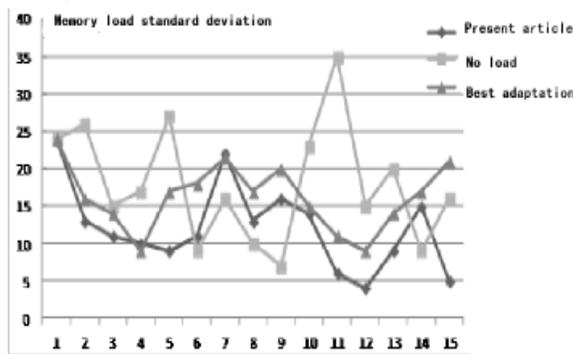


Figure.8 Memory load balancing

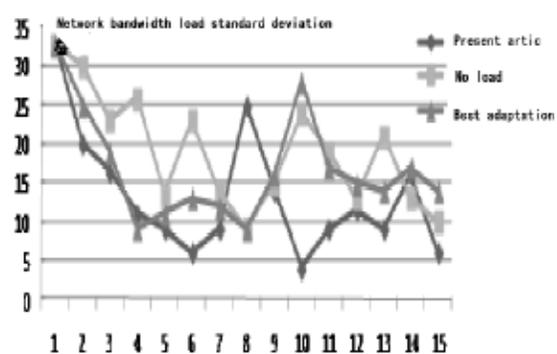


Figure.9 Network bandwidth load balance

We optimize and realize the design about load balance in cloud environment, and make relevant rules as far as platform completely modular, such as trigger rules, selection rules and location rules, and so on. The traditional methods based on the load balance system of cloud computing analyze anew the problem and improve it. The trigger rules is not simply based on threshold value, but on probability to predict. The selection and location rules both consider adequately CPU and memory usage, and analyze the best way of migration from overall consumption. The more available resource a node has, the easier the process that virtual machine is migrated to the most suitable destination node is. After the overall theoretical system is integrated, we conduct an experiment and analyze the result. Contrastive analyze CPU, memory, bandwidth, we can see the obvious performance advantages.

5. Summary and Outlook

The passage analyze concretely how load balance realize, design the concrete details of load balance module and has a further analysis that how to set up the frame of dynamic migration, and then set index of evaluating load balance. The characteristic of this aspect is the probability based, and solves the resource load in cloud environment, which is realized in the dynamic migration. The final result shows that algorithm fusion suggested has obvious performance advantages.

Now we just consider intensively the load balance of infrastructure later, and analyze concretely the location rules on load migration module. But about selection rules, we consider it from the ratio aspect, from consumption of overall migration, then from re-migrated aspect of available resource, and combine with the consumption about the increase of trigger rules to migration time. In addition, Information to initialize of algorithm is set by manual work. we collect load information with periodicity, and the periodicity is also set by manual work. We need to make sure the sampling period in the follow-up work.

References

- [1] Approach to cloud computing[M]. Beijing: Posts and telecom press, 2009:165.
- [2] Li yong: The study and analyze based on the dynamic migration technology of the virtual machine[Academic dissertation]. NUDT, 2007.
- [3] Li zhiwei, Wu qingbo, Tan yusong. The study of virtual machine dynamic migration based on the device agent machining. Application research of computers. The 26th volume, 2009.4.

- [4] Kaiqi Xiong, Harry Perros, Service Performance and Analysis in Cloud Computing, [C] In Proceeding of Congress on Services, July 2009: 693-700
- [5] Shi yangbin. A kind of load balancing algorithm based on the virtual machine live migration in cloud environment. [Master's thesis]. Shanghai: Fudan university, 2011.
- [6] Dina P, O Halloran D. The statistical properties of host load, In to appear in the Forth Workshop on Languages, Compilers, and Run-time Systems for Scalable Computers(LCR98) and CMU Tech[EB/OL].
- [7] Barnsley M. Fractals Everywhere[M]. NEW YORK: Academic Press, 1988:87.
- [8] Zhou wenyu, Chen huaping, Yang shoubao, Fangjun. The virtual machine cluster resource scheduling based on migration [J]. Journal of huazhong university of science and technology (JCR Science Edition), 2011, (39) SupplementI: 130-133.
- [9] Chang F, Dean J, Chomawat S. BigTable: A distributed storage system for structured data [J]. ACM Transactions on Computer Systems, 2008, 26(2): 1-26.
- [10] Chen guoliang, Sun guangzhong, Xu yun. Sabina chinensis integration research status and development trend of parallel computing [J]. Science, 2009, 54(8): 1043-1049.

Acknowledgements

This work was supported by the National natural Science Foundation of China (No.61472256, No. 61170277), Innovation Program of Shanghai Municipal Education Commission (No.12zz137), and the Huijiang Foundation (C14002).

Biographies

Sun Hong: female, Han, 1964-, from Beijing, China, Master, associate professor, School of Optical-Electrical and Computer Engineering University of Shanghai for Science and Technology, master tutor, associate professor direction of research; Business schools University of Shanghai for Science and Technology doctor graduate student; the main research direction: computer network communication and clouds computing, management science and engineering, Management Information and Decision Support System. Email: sunhong@usst.edu.cn, Telephone:13916902800

Wang Weifeng: male, Han, 1992-, master student, School of Optical-Electrical and Computer Engineering University of Shanghai for Science and Technology; the main research direction: cloud computing and management information system. Email: wwfhuo@163.com

Chen Shiping: male, Han, 1964-, from Zhejiang, China, professor, Ph.D. doctoral tutor Business schools University of Shanghai for Science and Technology, research direction: computer network communication and clouds computing, management science and engineering. Email: chensp@usst.edu.cn

Xu Liping: female, Han, 1986-, Master, associate professor, University of Shanghai for Science and Technology; the main research direction: cloud computing and management information system. Email: 5850487@qq.com