International Journal of Cloud Computing and Services Science (IJ-CLOSER)
Vol.3, No.1, February 2014, pp. 37~43
ISSN: 2089-3337

Survey on Load Balancing Through Virtual Machine Scheduling in Cloud Computing Environment

Vijaypal Singh Rathor*, R. K. Pateriya*, Rajeev Kumar Gupta*
* Departement of Computer Science & Engineering, MANIT

Article Info

Article history:

Received Nov 22th, 2013
Revised Jan 20th, 2014
Accepted Feb 26th, 2014

Keyword:

VM Scheduling
Load Balancing
Virtualization
Server Consolidation
VM Migration

ABSTRACT

In the cloud environment number of user can request for the services simultaneously. So there should be a mechanism that efficiently allocates the resources to the user, but resources in the cloud environment are highly dynamic and heterogeneous in nature. Because of this nature it is very difficult to fully utilize the resources with the proper resource balancing. In order to improving the system performance, resources must be properly assigned with minimum overhead time and load must be equally distributed on the physical machines. Proper VM scheduling can also reduce the number of migration that will increase the overall performance of the system. Numbers of VM scheduling methods have been proposed. This paper includes some exiting VM scheduling methodologies with their anomalies.

Corresponding Author:

Vijaypal Singh Rathor,
Department of Electrical and Computer Engineering,
Maulana Azad National Institute of Technology,
Bhopal, 462051, India.
Email: vijay.palrathor@gmail.com

1. INTRODUCTION

Cloud is a parallel and distributed computing system consisting of a collection of inter-connected and virtualized computers that are dynamically provisioned and presented as one or more unified computing resources based on service-level agreements (SLA) established through negotiation between the service provider and consumers[1]. Cloud computing is also called utility computing in which user accesses the resources as service via internet, and users doesn't know where these services are hosted [2]. In cloud computing resources are provided as services and these services are Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS), this is also known as a service delivery model [3]. Cloud computing can be deployed as a public cloud, private cloud and hybrid cloud and this is called as a deployment model model [3]. The figure 1 shows the cloud computing service delivery model with cloud deployment model.

Virtualization is the backbone of the cloud computing technology, which increase the resource utilization. Cloud computing is scalable and elastic in nature, so user resource requirements can be changes dynamically, so host may be overloaded or under loaded, which trigger the virtual machine migration. Efficient resource scheduling can be achieved through the virtualization that can provide the balance resource utilization and also can minimize the number of virtual machine migration. In cloud environment VM scheduling plays a vital role because cloud computing is a collection of heterogeneous resources that are distributed on different places, so there is need to properly assign user request to an appropriate physical machine with minimum overhead time and high resource utilization. Numbers of VM scheduling methods have been proposed in which different parameters are considered. VM scheduling can also be classified on the basis of resources requirement i.e. static and dynamic.

In this paper we are discussing some proposed VM scheduling method and their anomalies based on the parameters resource utilization, load balancing and server consolidation. Rest of this paper is organizing as follows section II explains the related work along with their anomalies and section III shows the comparative study and finally section IV shows the conclusion of this paper.

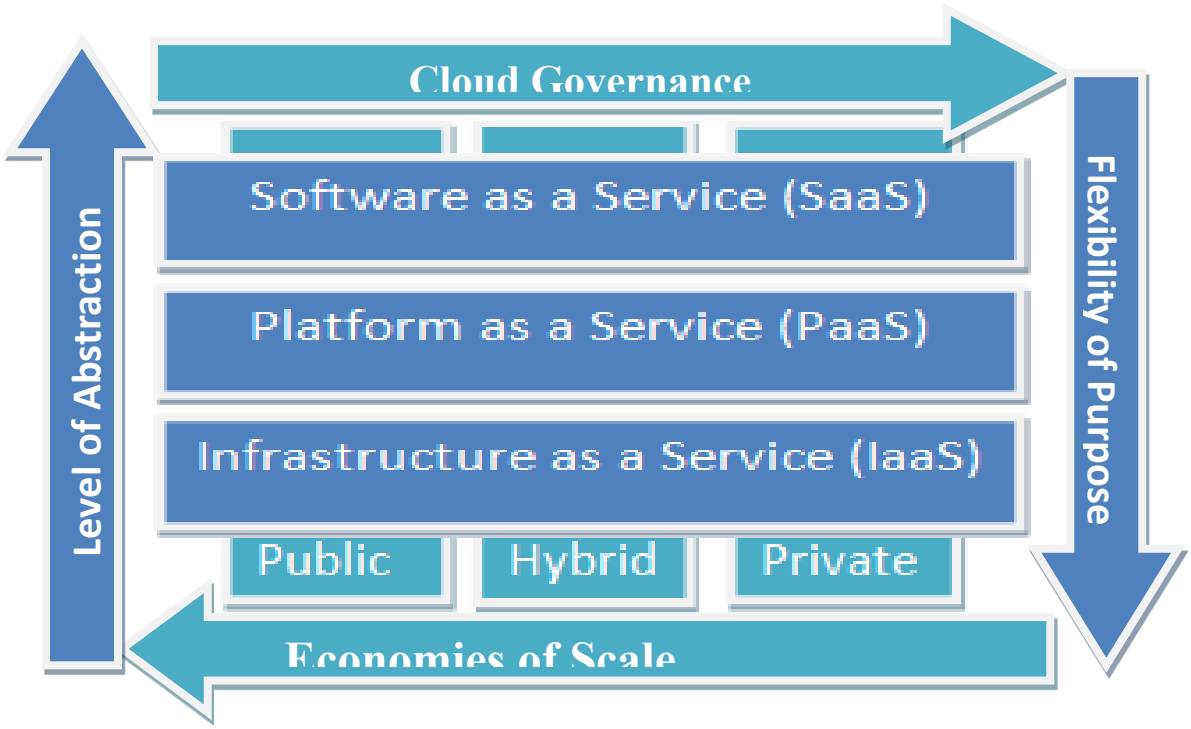

Figure 1. Cloud Computing Models

2. LITERATURE SURVEY

2.1. Optimized Control Strategy

Ke Yang et al. [4] proposed an Optimized Control Strategy for Load Balancing based on Live Migration of Virtual Machine which combines multi-strategy mechanism with the prediction mechanism. In this Strategy hosts are divided into four status domains (light-load domain, optimal domain, warning domain, overload domain) according to weighted average utilization of the CPU, memory, I/O and network bandwidth which is shown in figure 2. The weighted host utilization is calculated as

$$L_{host} = K_1 * L_{cpu} + K_2 * L_{mem} + K_3 * L_{net} + K_4 * L_{i/o} \quad (1)$$

$$\text{Where} \quad \sum_{i=1}^{4} K_i = 1 \quad (2)$$

Here L_{host} weighted host utilization which represents the loading condition of a host and L_{cpu}, L_{memory}, L_{net} and $L_{i/o}$ represents the CPU utilization, memory utilization, network bandwidth utilization and I/O utilization respectively of a host. According to the loading condition each host must lies in these following four domains.

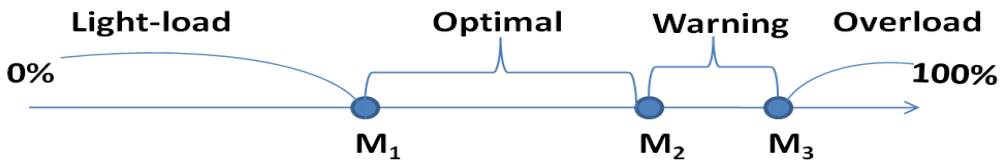

Figure 2. The four status domains

The hosts within different load status domain adopt different migration strategy based on the predicted future host utilization of resources using classical time series model [5], which consider migration timing, migration candidate VM and migration destination. Through this strategy, it reduces the number of the overloaded hosts, avoids instantaneous peak problem caused by the migration of virtual machines, solves the imbalance problem and the high cost problem in tradition scheduling algorithm of migration.

This method works as, when a host lies in optimal domain, if predicted load of a host is going up to warning threshold then choose biggest changed trend of resource (CPU or Memory or I/O)VM to migrate. The CPU trend is presented by the equation (3), similarly memory and I/O trend is calculated.

$$\mathrm{CPU}_{\mathrm{trend}} = \frac{\mathrm{CPU}_{\mathrm{future}}^{average} - \mathrm{CPU}_{\mathrm{now}}^{\mathrm{average}}}{\mathrm{CPU}_{\mathrm{now}}^{\mathrm{average}}} \qquad (3)$$

Where $CPU_{future}^{average}$ and $CPU_{now}^{average}$ are the average CPU utilization of future and current respectively.

When host lies in warning domain then firstly lock the host and prevent it from receiving the new tasks and judge that whether or not there is virtual machine being migrated on the host if there is anyone, we should wait; otherwise we should start the migration if host load goes down then unlock the host. When host a lies in overload domain, this host is overloaded, lock this host and highest weight VM should migrates.

The Optimized Control Strategy proposed the solution for the overloaded hosts and it doesn't consider the light load host, so this strategy can't provide server consolidation. This method uses the weighted average sum of CPU, Memory Network bandwidth and I/O for the loading condition but there may be a situation in which one of resource may exhausted and other will be in under utilize, so this situation can cause resource leak[6]. It requires O(n) destination searching time in bust case. There may be also possible that predicted load may not be same as the actual load.

2.2. Balanced Algorithm

Xin Li et al.[6] proposed a balanced algorithm for balancing resource utilization for continuous VM requests in cloud to avoid resource leak and reduce the energy consumption in the cloud datacenter. In this paper two approaches are proposed Greedy algorithm and balance algorithm to select the physical machine (PM) for virtual machine (VM) placement i.e. Greedy and Balanced. In Greedy approach hosts are divided into two categories i.e. Loaded and Standby, whereas in Balanced Algorithm the hosts divided into four categories. These categories are Resource Limited: The available resources on this PM are not sufficient to satisfy the new VM resource requirement, Fully Loaded: This PM will be fully loaded after satisfying the VM request, Resource-Leak: If this PM hosts the new VM request, resource utilization is abnormal which will cause resource leak. Standby: When the new VM is hosted on this PM still resource utilization will be regular. When a new request of VM placement comes, it only searches a standby PM that can host the VM. If there is no PM available in standby or in loaded, new PM is started to host the requested VM.

The Greedy approach is simple but resource leak occurs, a Balance Algorithm is proposed to avoid resource leak so that energy consumption can be reduced by minimizing the number of PMs. In this Li states that power consumption is very similar though the CPU utilization varies, so long as the PM is active, so Li calculate the power consumption for static VM placement as

$$\text{Power} = \alpha * \text{m} \qquad (1)$$

Where m is number of host running and α is the power consumption factor of a host in a time slot. But in case of continues VM placement the number of hosts are varies as requests are reached, so power consumption for continues VM placement can be calculated as

$$Power = \sum_{i=1}^{n} \alpha * m_i \qquad (2)$$

Where m_i is the number of running hosts when i virtual machine requests have reached the cloud and n is the number of virtual machine requests.

If there are n number of hosts, to find out the Standby host the Balance algorithm can take O(n) searching time in bust case. In the Balance algorithm first places the VM on the Fully Loaded PM but resources requirement can change dynamically, so if resources requirement will be increases, result in the instant VM migration, due to this system performance can be degraded.

2.3. Dynamic And Integrated Resource Scheduling (DAIRS) Algorithm

Wenhong Tian et al.[7] proposed a dynamic and integrated resource scheduling (DAIRS) algorithm for Cloud datacenters which develops integrated measurement for the total imbalance level of a cloud datacenter and average imbalance level of each physical machine. To calculate the total imbalance level of a cloud datacenter and average imbalance level of each physical machine DAIRS defined the following parameters.

The average utilization of all CPUs in a Cloud datacenter, is defined as

$$CPU_u^A = \frac{\sum_i^N CPU_i^U CPU_i^n}{\sum_i^N CPU_i^n} \quad (1)$$

Where CPU_i^U is an averaged CPU utilization during observed period of a single server i, CPU_i^n be the total number CPUs of server i and N is the total number physical servers in cloud datacenter. Similarly memory and network bandwidth utilizations also are calculated. And then integrated load imbalance value ILB_i of server i is defined as

$$\frac{(Avg_i - CPU_u^A)^2 + (Avg_i - MEM_u^A)^2 + (Avg_i - NET_u^A)^2}{3} \quad (2)$$

$$Where \quad Avg_i = (CPU_i^U + MEM_i^U + NET_i^U)/3 \quad (3)$$

The imbalance value of all CPUs in a data center using absolute deviation, is calculated as

$$ILB_{CPU} = \sum | CPU_i^U - CPU_U^A | \quad (4)$$

The ILB_{MEM} and ILB_{NET} can be calculated in similar fashion. Then total imbalance values of all servers in a Cloud datacenter are given by

$$ILB_{tot} = \sum_i^N ILB_i \quad (5)$$

The average imbalance value of a physical server i is calculated as

$$ILB_{Avg}^{tot} = \frac{ILB_{tot}}{N} \quad (6)$$

And the average imbalance value of a Cloud datacenter (CDC) is calculated as

$$ILB_{Avg}^{CDC} = \frac{ILB_{CPU} + ILB_{MEM} + ILB_{NET}}{N} \quad (7)$$

The equation (6) and equation (7) are used to measure the degree of overload of a host and of the system respectively and the allocation is also based on this degree. In the allocation algorithm the physical machines are sorted in ascending order according to the resource utilization and divides physical machines into multiple intervals of a particular size. When request of VM placement comes then select the PM with lowest utilization interval (for example (0,0.10)) and start virtual machine allocation, as long as the allocation of the virtual machine does not exceed the maximum capacity of that physical machine. If any host resource utilization exceeds the preset utilization threshold, the virtual machines with lowest load should be migrated using allocation algorithm until the utilization of PM is under threshold.

In allocation VM is always placed on lowest utilization interval PM, so it means it uses the worst fit concept, result in low resource utilization and energy consumption increases.

2.4. Modification Best Fit Decreasing Algorithm

Rajkumar Buyya et al. [8] proposed a Modification Best Fit Decreasing (MBFD) algorithm for energy-efficient resource allocation to provide server consolidation, in which VM allocation problem is divided into two parts, the first part consider the creation of new VMs for user requests and assigning them on to the physical machines while on the other hand second part optimized the current allocation of VMs.

To solve the VM allocation problem, modification of the Best Fit Decreasing (BFD) algorithm [9] is proposed in which VMs are sorted in decreasing order of current utilization and allocate each VM to a host so that it provides the least increase of power consumption. And to optimize the current allocation the upper and lower utilization threshold are used. If host utilization crosses the upper utilization threshold,

migrate some of VMs from this host to prevent the SLA. If host utilization goes below the lower threshold, migrate all VMs from this host and switch off it to minimize the energy consumption, which is known as a server consolidation. To calculate the total energy consumption in cloud Buyya et al. defined the power model P (u) as

$$P(u) = k * P_{max} + (1-k) * P_{max} * u. \quad (1)$$

Where P_{max} is the maximum power consumed when the server is fully utilized; k is the fraction of power consumed by the idle server; and u is the CPU utilization which is also a function of time because it can change over the time due to variability of the workload, so can be represented as u(t), and then total energy (E) consumption can be defined as

$$E = \int_t P(u(t)) \quad (2)$$

If there are m number of hosts and n VMs that have to allocated then the time complexity of MBFD allocation algorithm is n*m. And the complexity of the proposed Minimization of Migration (MM) algorithm is proportional to the product of the number of over- and under-utilized hosts and the number of VMs allocated to these hosts. MBFD used the fixed value for threshold but it is unsuitable for dynamic and unpredictable workloads environment, in which different types of applications can share physical resources. To solve this problem Adaptive Threshold-Based Approach is proposed [10].

2.5. Novel Vector Based Approach

Mayank Mishra et al.[11] proposed a novel Vector Based Approach for VM placement. This approach based on 3-D vector in which each dimension of a vector denotes one type of resource (i.e. CPU, MEM, I/O). The primary goal of this algorithm is to make the resource utilization of PMs as balanced (along each resource dimension) as possible.

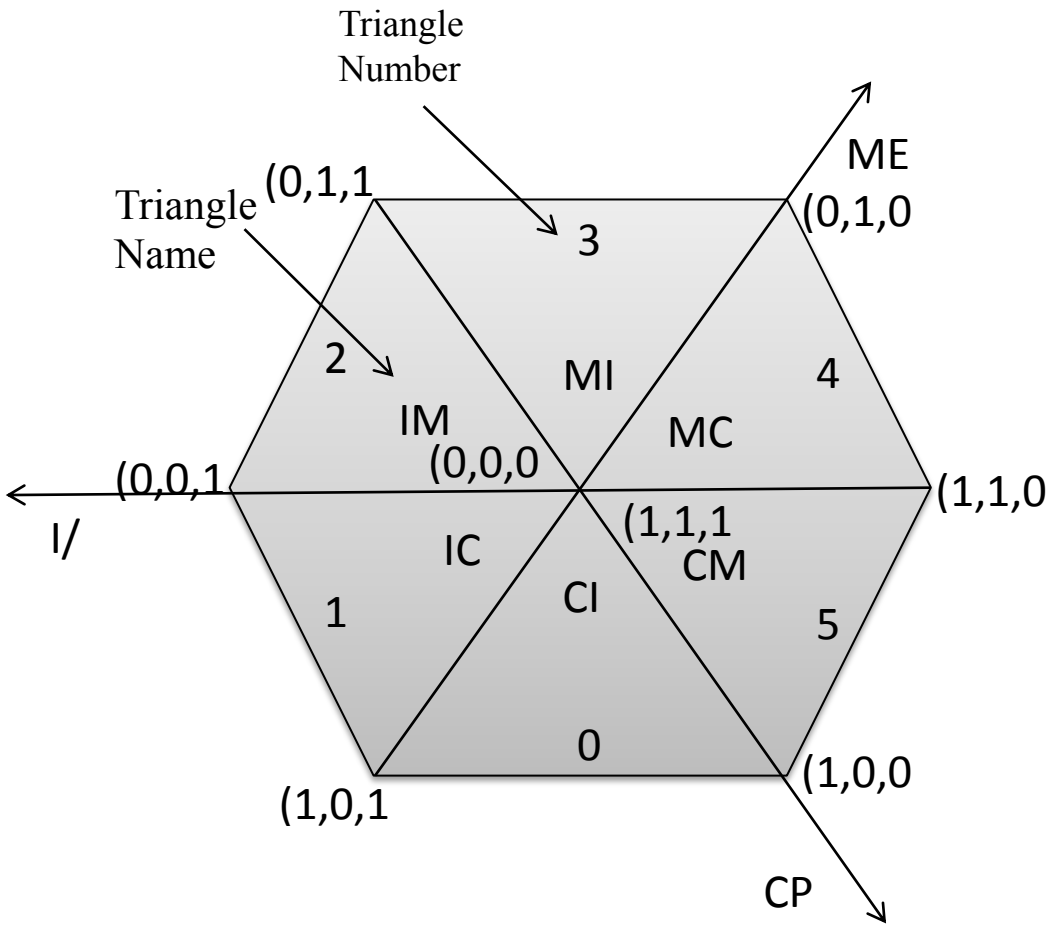

Fig. 3 The Planar Resource Hexagon (PRH)

This would require that we have a way of finding complementary destination PM to place a VM, means a VM which has less CPU requirement than MEM requirement should be placed on a PM which has greater CPU utilization than MEM utilization. To find the complementary PM for a VM a planer resource hexagon is obtained by projecting a cube, in which each triangle denotes a PM and the opposite triangles of planer resource hexagon are complementary to each other which is shown in figure 3 in which triangle number 3 and name MI (i.e. M>I>C) and triangle number 0 and name CI (i.e. C>I>M) are complementary to each other.

This methodology provides the efficient and balance resource utilization. It increases the resource utilization and reduces the energy consumption by server consolidation in cloud datacenter. This approach also has good features over the previous placements methods SandPiper [12] and VectorDot [13]

The novel approach is based on the 3-D vector in which each dimension of a vector denotes the one resource type i.e. CPU, MEM, I/O. And if resource dimensions will be more than three, this method can't work anymore.

In this approach if a virtual machine has unequal resource requirement in all dimension or has only less than or greater than relationship (i.e. C>M>I or C>I>M or M>I>C or M>C>I or I>C>M or I>M>C) between the resources requirement, the complementary physical machine can be find easily. It means a virtual machine which has resources requirement as CPU>MEM>IO should be placed on the physical machine which has resources utilization as IO>MEM>CPU. But if a virtual machine which has equal resources requirement in two or more dimensions or has a equal relationship (i.e. C=M<I or C=I<M or I=M<C or I=m=C etc.) between the resources requirement could not have any complementary physical machine . For example a virtual machine which has a resources requirement CPU=MEM>IO can't have a complementary physical machine like IO>MEM=CPU.

3. COMPARATIVE STUDY

The comparative study is shown in tabular form. In table various above discussed methods are compared based on the some parameters which are considered by any of these algorithms. The results are displaying as YES or NO of algorithms versus parameters.

Table I. Comparative study of above methods

Parameters / **Algorithm**	Provide Balance Resource Utilization	Reduces the Energy Consumption	Minimizing Response Time	Work For Resources Type >3
Optimized Control Strategy	NO	NO	NO	YES
Balanced Algorithm	YES	YES	NO	YES
DAIRS	YES	NO	YES	YES
MBFD	NO	YES	NO	YES
Novel Approach	YES	YES	YES	NO

4. CONCLUSION

Cloud computing is a collection of large number of heterogeneous interconnected computers and resources requirement can changes dynamically, so assignment of virtual machine on a suitable host is a very challenging task. In this paper different VM allocation methods are discussed based on some parameters (i.e. resource utilization, server consolidation and complexity etc.) with their anomalies, so there is a need of a VM scheduling method which will contain the best feature of these algorithms and also remove anomalies present in above discussed methodologies.

REFERENCES

[1] *R. Buyya, J. Broberg, A. Goscinski*, "Cloud Computing : Principle and Paradigms", 1st ed., Hoboken: John Wiley & Sons, 2011.

[2] *A. Weiss.* Computing in the Clouds. netWorker, 11(4): 16-25, ACM Press, New York, USA, Dec. 2007.

[3] *T. Mather, S. Kumaraswamy, and S. Latif,* "Cloud Security and Privacy", 1st ed., USA : O'Reilly Media, 2009, pp. 11-25.

[4] *K. Yang, J. Gu,T. Zhao and G. Sun* , "An Optimized Control Strategy for Load Balancing based on Live Migration of Virtual Machine ", in Proc. 6th Annual Chinagrid Conference (ChinaGrid), Liaoning : IEEE, 2011.

[5] *W. Gersch and T. Brotherton,* "AR model prediction of time series with trends and seasonalities: A contrast with Box-Jenkins modeling", in Proc. 19th IEEE Conference on Decision and Control including the Symposium on Adaptive Processes,1980,19(1): 988-990.

[6] *X. Li, Z. Qian, R. Chi, B. Zhang, and S. Lu,* "Balancing Resource Utilization for Continuous Virtual Machine Requests in Clouds", in Proc. Sixth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), Palermo: IEEE, 2012.

[7] *W. Tian, Y. Zhao, Y. Zhong, M. Xu and C. Jing,* "A dynamic and integrated load-balancing scheduling algorithm for Cloud datacenters", in Proc. International Conference on Cloud Computing and Intelligence Systems (CCIS), Beijing : IEEE, 2011.

[8] *R. Buyya, A. Beloglazov, and J. Abawajy*, "Energy-Efficient Management of Data Center Resources for Cloud Computing: A Vision, Architectural Elements, and Open Challenges", in proceedings of the 2010 International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA 2010), Las Vegas, USA, July 12-15, 2010.

[9] *M. Yue.* "A simple proof of the inequality FFD (L)< 11/9 OPT (L)+ 1,for all l for the FFD bin-packing algorithm". Acta Mathematicae Applicatae Sinica (English Series), 7(4):321331, 1991.

[10] *A. Beloglazov and R. Buyya,* "Adaptive Threshold-Based Approach for Energy-Efficient Consolidation of Virtual Machines in Cloud Data Centers", in Proc. 8th International Workshop on Middleware for Grids, Clouds and e-Science, Ney York : ACM, 2010.

[11] *M. Mishra and A. Sahoo,* "On Theory of VM Placement: Anomalies in Existing Methodologies and Their Mitigation Using a Novel Vector Based Approach", in Proc. IEEE International Conference on Cloud Computing (CLOUD), Washington: IEEE, 2011.

[12] *T.WOOD, P. Shenoy and A. Venkataramani*, "Black-box and Gray-box Strategies for Virtual Machine Migration", in the proceedings 4th USENIX conference on Networked systems design & implementation(NSDI), Berkeley : ACM, 2007.

[13] *H. ZHENG, L. ZHOU, J. WU,* "Design and Implementation of Load Balancing in Web Server Cluster System", Journal of Nanjing University of Aeronautics & Astronautics, Vol. 38 No. 3 Jun. 2006.