CLUSTERING BASED TASK DEPLOYMENT APPROACH FOR LOAD BALANCING USING BAYS THEOREM IN CLOUD COMPUTING

Girish U.Pimpale[#1]
[#1]PG Student, Department of Computer Engineering, SSVPS'S B.S.Deore College of Engineering, India

Abstract- Cloud Computing has Large Scale Distributed Infrastructure which provide the services like infrastructure and software and platform. Cloud computing provides on demand service in which the user has to pay for the services he uses.Focusing at the current problems that most physical hosts in the cloud data centre are so overloaded that it makes the whole cloud data centre load imbalanced and that existing load balancing approaches have relatively high complexity, the proposed system focuses on the selection problem of physical hosts for deploying requested tasks. The concept of achieving the overall load balancing in a long-term process in contrast to the immediate load balancing approaches CB-TD makes a limited constraint about all physical hosts aiming to achieve a task deployment approach with global search capability in terms of the performance function of computing resource. The Bayes theorem is use here with the clustering process to find out the optimal clustering set of physical hosts.

Keywords: Task Deployment, Load Balancing, Bayes Theorem, Clustering, Cloud Computing

I. INTRODUCTION

To achieve flexible and on demand services for number of companies, cloud computing becomes more attractive. Cloud computing can be a type of parallel and distributed system which consist of collection of inter connected and virtualized computers that are dynamically provisioned and represented as one or more computing resources based on service level agreements which are established between service provider and customers the satisfactory service performance is easy to achieve immediately since the available amount of computing resource in cloud date centre is always larger than the requested amount though this kind of Requirements are real-time and strict. Also, as the total resource amount requested by all users accumulated within a processing cycle has rarely possibility to approach the current total amount of available computing resource in a cloud data center, the availability and performance potential of the cloud data centre don't need to keep being maximized at all times based on the points above, it is unnecessary to ensure that the optimal load balancing effect is reached through after each algorithm cycle in real time as long as the whole system has been always tending to a long-term optimal load balancing effect. Respectively data pre-processing, pattern discovery, and pattern analysis. In data preprocessing raw data is filtered to get required data, then patterns, rules and statistics are figure out so that useful rules and patterns are determined in pattern analysis phase. These interesting patterns are very useful to predict web users browsing behavior.

To find out the optimal physical hosts for tasks deployment by achieving a load balancing strategy through a process and thus to obtain optimal performance. A constraint value is determined in the light of the resource amount of requested tasks. That is, each requested task has a constraint value. Then the physical hosts, whose constraint values of computing resource are greater than the constraint value of the tasks in the cloud data center, are clustered. And the physical hosts whose similarities are within a certain threshold value constitute a set through clustering. And the physical host set obtained by clustering methods is the physical host set whose physical hosts are optimal for deploying the task. Then, the tasks will be put into physical hosts in the set to deploy. The process of the clustering of physical hosts in the cloud data center is to find the optimal physical hosts for deploying tasks. Thus, through the task deployment strategy, not only the load balancing of the cloud data center can be achieved, but also efficient performance of external services can be provided for users.

The organization of this paper is as follows. In Section II, we present the work related to the different prediction models. Section III describes the analysis of the limitations of the existing systems. The working of the proposed system is discussed in section IV. Section V discusses the details of experiment with results. Finally in section VI we conclude and discuss future scope.

II. RELATED WORK

In 2011 Vivek Shrivastava [13] have focus on this problem an approach to place the virtual machines with strong correlation of applications intensively. However, he was unable to focus on the load balancing problem and cost overheads of clouddata centers. He was only focused on virtual machines management to strengthen the managementof cloud data centers and increased the performing efficiency of Iaas cloud data centers.

In 2012 MahfuzurRahman [14]and Peter Graham have proposed a hybrid approach which combines static and dynamic provisioning. He uses to adapt a good initial static placement of virtual machines in response to involve load characteristics using actual migration of virtual machine to increase the computing efficiency of cloud data centers.

Corentin Dupont In 2013[15] had proposed a flexible solution of reallocation of virtual machines in a cloud data center thus to calculate and work out the effective placement of virtual machines to obtain the overall load balancing of cloud data centers.In addition, Jia Zhao[16]in 2013 have proposed an approach which is a heuristic and self-adaptive multi objective optimization method based on the improved genetic algorithm (GA) and the theory of Pareto optimal solutions.It target is to achieve dynamic load balancing.

In 2013, Jing Tai Piao[17]had proposed a solution of network-aware virtual machine placement and migration approach to minimizing the data transfer time consumption and it improve the overall application performance of the cloud data center. The utilization of resource of physical hosts and raises the operational cost of the cloud data center. Jason Sonneck focus on this problem and presented a solution of decentralized affinity-aware migration technique to increase incorporates diversity and dynamism in network topology and job communication patterns to assign virtual machines on the available physical resources. The solution monitors network affinity between pairs of virtual machines and uses a distributed bartering methods coupled with migration, to dynamically adjust virtual machine placement such that communication overhead is minimized and load balancing is achieved. Sau-Ming Lau also addresses this problem and proposed a solution of having integrated the two strategies of heavy load priority and light loadpriority. The proposed solution is an adaptive load distribution algorithm to effectively reduce communication overhead of the load balancing process. Utilizing the greedy process can solve the problem of load. However, many algorithms above cannot meet greedy choice performance and the nature of optimal sub-structure at the same time.

III. ANALYSIS OF PROBLEM

In IaaS,Paas,Saas cloud data centres, the system will deploy tasks on the physical hosts in the resource pool of the cloud data center when users submit task requests. In general, the cloud data center will choose the physical hosts at random to deploy tasks. When the resource amount requested by a task is greater than the remaining resource amount of the physical host, the physical host are not able to deploy the task. And also when the requested resource amount is close to the remaining amount of the physical host, the physical host will have some heavy workload, causing the decline of its service capacity and computing power. The situation leads to load imbalance of the cloud data center and make service efficiency fall.

IV.PROPOSED WORK

This section will describe processes for Load balancing using bays therom for load balancing which is used to deploy task requests received by the cloud data center into optimal target physical hosts in the IaaS cloud computing data center. Its process is what combines Bayes theorem with clustering. It has achieved the overall load balancing of the entire network from the perspective of cloud data centers' Figure 1 shows the working flow of the proposed system.

Step 1

The proposed System approach is a new way of task deployment approach, which is used to deploy task requests received by the cloud data center into optimal physical hosts in the IaaS cloud computing data center. Its algorithm flowchart is what uses Bayes theorem with clustering. It has achieved the overall load balancing of the entire network from the perspective of cloud data centers' long-term operations and thus to improve the performance and efficiency.

First, these physical hosts, each of which has a larger remaining resource amount than the maximum requested resource amount of all task requests, can be searched out to constitute a new candidate set to meet the performance constraint while making CB-TD have the potential of achieving the long-term load balancing. Second, the *n*physical hosts in the set of physical hosts can be regarded as *n* objects waiting for being clustered. Each physical host in the given set is given to a prior probability. The posterior probability of each physical host's handling tasks can be calculated through Bayes theorem. This probability can be regarded as an attribute of each object while the remaining CPU.

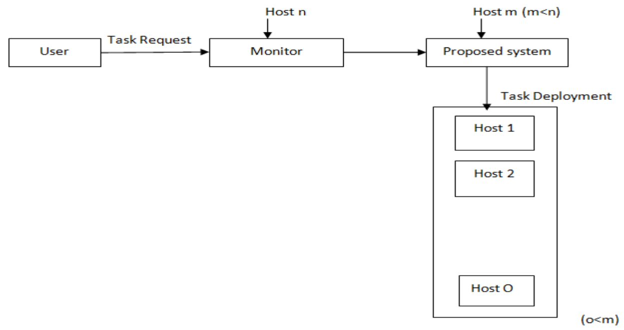

Fig.1 System Work Flow

Resource amount and the remaining memory resource amount of each physical host can be regarded as the other two attributes. The similarity degree values between physical hosts are calculated according to the three attributes of each physical host. A threshold value is determined on the basis of these similarity degree values. The physical hosts whose similarity degree values between them are within the given threshold can be seen as the optimal clustering to form the final set of candidate physical hosts. Finally, the tasks are placed on the hosts in the final set. And the clustering process of physical hosts in the cloud data centre is the process of finding the optimal physical hosts for executing tasks.

The Working flow of an algorithm which explains the how the task is going to deploy on the host according to their resource requirements and finally it achieve the overall load balancing effect.

Step 2:

In IaaS cloud data centers, there are too large numbers of physical hosts. In order to avoid this situation that the selected physical hosts are not able to meet the resource requirements of requested tasks and to achieve the best clustering effect through minimizing the candidate set, system assume that there are *m* physical hosts in the cloud data center, and each physical host needs to be assigned to a constraint value for measuring its remaining available computing power in the cloud data center.After that the actual resource requirement of each physical host is calculated and if the resource amount requested by task is able to fulfil by the host then it is added to new host list further deployment of task and if it is not then it will be discarded. The maximum requested resource amountof host intask requestcan be calculatedAnd if it is greater thathost will be placed into the new set of physical hostHaving compared the constraintvalueof each physical host with the performance constraint maximum valuethe new candidate setis obtained and it will be used as the candidate set of physical hosts for the clustering process.

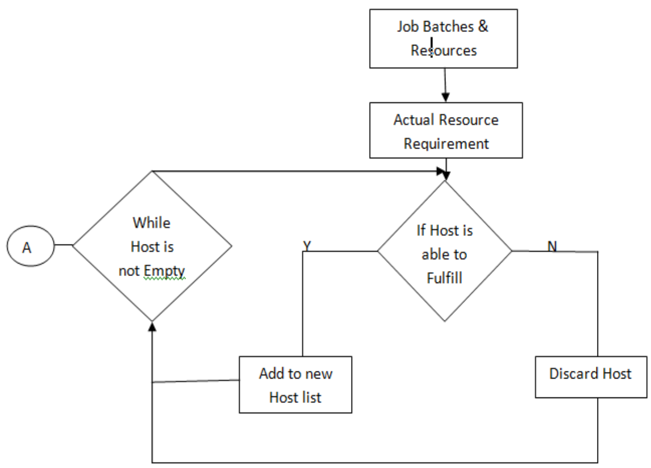

Fig.2 Flowchart of proposed system (Part a)

Step 3:

These physical hosts with better performance in the cloud data centre are put into the set of physical host by the restricting of the constraint value of the requested tasks. However, the system cannot ensure that each physical host from the set of physical hostcan process any of the received task request in task requestsince some physical hosts may have large amount of available memory resource .The memory values are also very large, but they aren't what system want. Aiming to not only fully exploit the advantage of the weighted sum of multiple kinds of resource but also find out its disadvantage that there exist some unreasonable hosts, the system utilized the methods of probability theory and clustering to obtain the selection of optimal hosts while achieving overall long-term load balancing. It is by picking out the optimal clustering of physical hosts with the relatively larger computing power to process these received tasks in set of task request that the objective of load balancing is achieved from a long-term perspective targeting to achieve a better clustering effect, the posterior probability of each physical host is introduced into the proposed approach and used as an attribute of each object.

For calculating Posterior probability

$$P(B_i|A) = \frac{P(A|B_i) * P(B_i)}{\sum_{i=1}^{m'} P(A|B_i) * P(B_i)}$$

Where Event A is defined as that those tasks are executed on some physical hosts. And event *Bi* is defined as the event that the physical host *i* is chosen.

The posterior probability value $P(Bi|A)$ of each physical host *i* in the new host list has been obtained. The posterior probability values of the physical hosts in the set new physical host are sorted descending. The new host listrepresents the physical host with the biggest posterior probability in old host listand thus new host list is selected as the clustering centre.

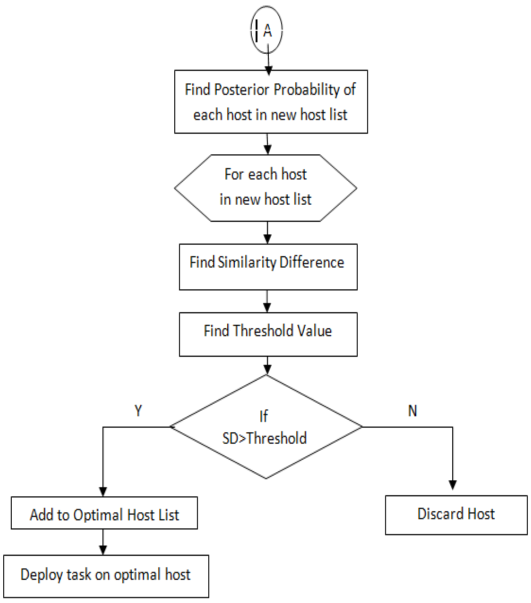

Fig.3 Flowchart of proposed system (Part b)

Step 4:

In next step similarity difference is find out between the old host list and the new host list which contain the host with maximum posterior probability by using the formula

$$SD = \frac{1}{\sqrt{(P_i^1 - P_j^1)^2 + (L_{ci}^2 - L_{cj}^2)^2 + (L_{mi}^3 - L_{mj}^3)^2}}$$

Where P1 and P2 are posterior probability, L1 and L2 are the host list interms of CPU and L3 and L4 are host list interms of memory of old host list set and new host list respectively. After that similarity threshold value is find out among all this similarity differences value. if the similarity degree value is greater than similarity threshold value then that host is place in new host list 3 which will be the final set of host list on which the task are going to deploy.

Step 5:

Repeat the above process the process of calculating physical hosts posterior probability is the process of applying Bayes theorem. The reason why the process of calculating posterior probability values in proposed system is in line with that of Bayes theorems application is that Bayes theorem provides effective means to amend the original judgment by using the collected information.

VI-EXPERIMENTAL RESULTS

In this section, the DLB deployment approach and the proposed deployment approach are compared as

Following aspects:

1) Make Span
2) Standard deviation values measuring load balancing Effect
3) Throughput measuring external service performance
4) Failure number of task deployment events
5) Incremental percentage of their standard deviation Values

Cloudsim is a new open source toolkit developed using java that generalized, and advanced simulation framework allows simulation of Cloud computing and application services. Cloudsim is a simulation tool for creating cloud computing environment and used as the simulator in solving the Load balancing problem. Cloudsim allows us to create a data center with a set of hosts and number of virtual machines as resources.

Virtual Machine - It is implemented virtual software of a computer that executes application programs same as a physical machine.

Cloudlet - Cloudlet is input job or set of tasks to be executed in cloud environment. Cloudlet has its own unique Cloudlet_id, and Cloudlet_length.

In the Cloudsim platform, this paper has compared the proposed deployment strategy with the dynamic load balancing strategy (DLB). They are evaluated and tested through five different sets of simulation experiments. The final simulation results have shown that the proposed approach can not only make the failure number of deployment tasks lower than that of DLB, but also make the cloud data centre have a better load balancing effect. Compared with other deployment approaches, proposed system has a better stability and efficiency in the aspect of achieving the overall load balancing of cloud data centres, especially for deploying a large amount of continuous task requests.

Comparison on Make span

In the experimental scenario, CB-TD is compared with the DLB approach on MakeSpan, which is the needed time of processing tasks set. The experimental results of the twoapproaches are shown as in respective figure. The MakeSpan values of requested tasks set increase along with the number of the requested tasks increasing. The other approach has deployed the requested tasks at random on the physical hosts in the cloud data center essentially. On this occasion, with the increase of the number of requested tasks, the ability of handling tasks will weaken gradually. Therefore, the needed time will also increase for sure. And the DLB deployment approach just predicts requested tasks' resource amount on the basis of its knowledge repository in the beginning and will trigger task deployment events according to the revenue value of load balancing. With the increase of the number of requested tasks, the communication cost between physical hosts will increase surely. The ability of handling tasks will weaken gradually, and the time needed will increase. However, it is smaller than that of the DLB approach. The proposed approach will select the optimal physical hosts set to deal with tasks in each round of iterations, and deploy tasks into the corresponding hosts. It not only avoids a large amount of communication cost, but also guarantees physical hosts computing performance. The time of handling tasks will increase with the increase of the number of requested tasks. It is smaller than that of the DLB deployment approach with the same number of requested tasks. Compared with the DLB approach, CB-TD has smaller MakeSpan value in the same condition, which can be shown from Figure 3. This experiment illustrates that CB-TD has not only a better load balancing effect but also a better MakeSpan value relatively.

Table 1.Makespan Comparison Value

No.of Task	CB-TD approach time (s)	DLB approach time(s)
5	287.68	302.52
10	513.83	525.64
15	990.83	1005.45
20	1151.69	1197.63
25	1664.76	1701.85

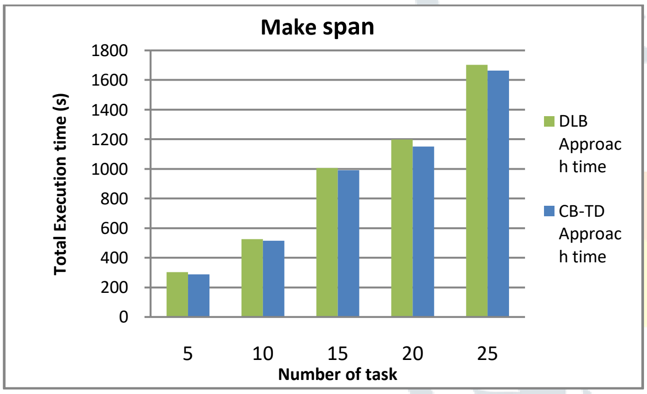

Fig 4.Comparision on Makespan

Comparison on Standered Deviation value

In this set of experiments, the changes of DLB and CB-TD's load balancing effect in the cloud data center with time changing are compared. Here, we have employed the standard deviation value indicated above and used to measure the degree of load balancing to conduct the set of experiments. Obviously, a smaller standard deviation value represents that the cloud data center has the better balancing of load. With the increase of the time for the deployment process in the cloud data centre, According to the results of this set of experiments, the proposed system approach has a better effect of load balancing and thus to improve the resource utilization of the cloud data center effectively.

Table 2 .Standered Deviation Value

No. of Task	CB-TDStd. Deviation value	DLB Std. Deviation value
5	4.6	5.4
10	4.9	5.9
15	5.6	6.2
20	5.8	6.4
25	5.9	6.8

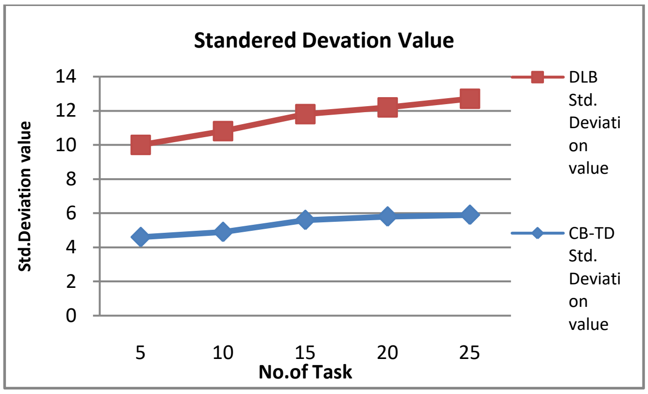

Fig.5 Comparision on Standered Deviation

Comparison on Throughput value

In the experiment scenario, the proposed CB-TD approach is verified by comparing the external service performance of the cloud data centre respectively implementing the three deployment approaches with time increasing. The throughput can represent the comprehensive evaluation of cloud systems, such as components ability of dealing with tasks, the transmitting ability of data and the ability of responding task requests to users etc and it can be calculated as the number of request fetch per second. However, with time increasing its external service performance is not stable and has awaking mode. Also, the external service performance of CB-TD is relatively better than that of DLB at all the time.

Time	CB-TD Throughput	DLB Throughput
200	0.5	0.8
400	0.9	1.4
600	1.4	2.1
800	2	2.7
1000	2.7	3.4

Table 3.Throughput Values

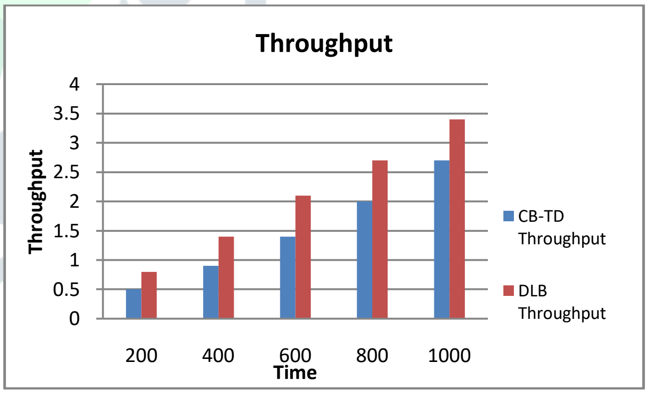

Fig 6.Comparison on Throughput

Comparison on failure no. of task :

In this experiment the proposed CB-TD approach with the DLB approach on the failure number of deploying tasks.

No.of Task	CB-TD Failure no.of Task	DLB Failure no.of Task
5	1	10
10	2	15
15	14	25
20	16	28
25	19	30

Table 4.No.of Failure task

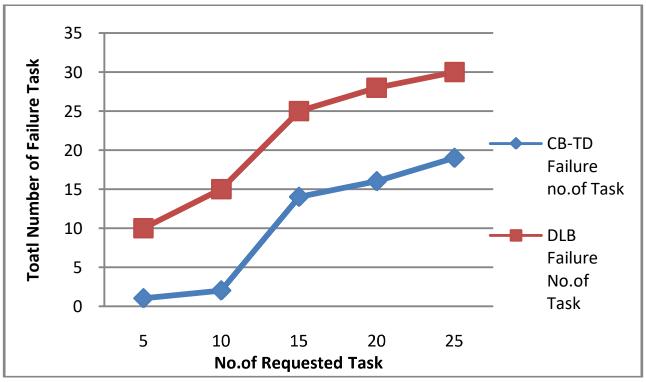

Fig.7. Comparison on Failure Number of Task

As shown in Figure 8, in the simulated cloud data center, with the increasing of requested tasks the failure numbers of DLB are more and more. However, in the cloud data center implementing proposed system with the requested tasks increasing, the failure number of deploying tasks is increasing very slowly and is always lower than those DLB

Comparison on Incremental Percentage of Standard Deviation Value:

In this experimental scenario, with the number oftask requests increasing, it was verified the loadbalancing effect of DLB and CB-TD by comparing their percentages of the incremental standard deviation values

Table 5.Values for Incremental Percentage for Std. Deviation

No.of Task	CB-TD	DLB
5	4.6	5.4
10	4.7	5.9
15	5.3	6.2
20	5.5	6.4
25	5.6	6.8

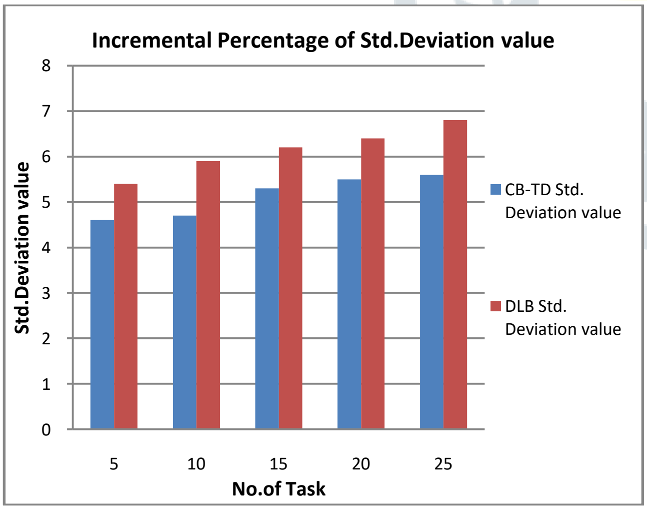

Fig.8. Comparison on Incremental Percentage of Standard deviation value

As illustrated in Figure 8, with the increase of the requested tasks, the percentage of load in the cloud data center increases and the load balancing effect of CB-TD is better than that of DLB.The Proposed System has ability to deploy every requested task on the optimal physical host quickly and to ensure the overall load balancing of the cloud data centre from a long-term perspective.

VII. CONCLUSION

Based on concluding the related work, this task proposes a new load balancing strategy,CB-TD which is based on task deployment and gives the main idea including process implementation and evaluation. It uses methods, ideas, which are based on Bayes theorem and the clustering process. The proposed system first has narrowed down the search scope by comparing performance values. Then it has utilized Bayes theorem to obtain the posterior probability values of all candidate physical hosts. Finally it has combined probability theorem and the clustering idea to pick out the optimal hosts set, where these physical hosts have the most remaining computing power currently, for assigning and executing tasks by selecting the physical host with the maximum posteriori probability value as the clustering center and thus to achieve the load balancing effect from the long-term perspective. Simulation experiments demonstrate that the proposed CB-TD approach can deploy the instant tasks quickly and effectively in cloud data centers. It makes cloud data centers achieve a long-term load balancing of the whole network.

REFERENCES

[1] S. K. Garg, S. Versteeg, and R. Buyya, ”A Framework for Ranking of Cloud Computing Services,” Future Generation Computer Systems, vol. 29, no. 4, pp. 1012-1023, Jun. 2012.

[2] G. Nan, Z. Mao, M. Li, Y. Zhang, S. Gjessing, H. Wang, and Mohsen Guizani, ”Distributed Resource Allocation in Cloudbased Wireless Multimedia Social Networks,” IEEE Network Magazine, vol. 28, no. 4, pp. 74-80, Jul. 2014.

[3] R. Yu, Y. Zhang, S. Gjessing, W. Xia, and K. Yang, ”Toward Cloud-based Vehicular Networks with Efficient Resource Management,” IEEE Network Magazine, vol. 27, no. 5, pp. 48-55, 2013.

[4] G. Xu, Y. Ding, J. Zhao, L. Hu, and X. Fu, ”A Novel ArtificialBee Colony Approach of Live Virtual Machine Migration PolicyUsing Bayes Theorem,” The Scientific World Journal, vol. 2013,no. 2013, p. 369209, Sept. 2013.

[5] J. Zhao, L. Hu, Y. Ding, G. Xu, and M. Hu, ”A HeuristicPlacement Selection of Live Virtual Machine Migration forEnergy-Saving in Cloud Computing Environment,” PloS one,vol. 9, no. 9, p. e108275, Sept. 2014.

[6] L. Liu, H. Wang, X. Liu, X. Jin, W. He, Q. Wang, and Y. Chen,”GreenCloud: A New Architecture for Green Data Center,”Proc. the 6th International Conference Industry Session onAutonomic Computing and Communications Industry Session,pp. 29-38, Jun. 2009.

[7] Q.Wei, G. Xu, and Y. Li, ”Research on Cluster and Load Balancebased on Linux Virtual Server,” Proc. Information Computingand Applications, pp. 169-176, 2011.

[8] Y. Di, S. Wang, and X. Sun, ”A dynamic load balancing modelbased on negative feedback and exponential smoothing estimation,”Proc. the 8th International Conference on Autonomic andAutonomous Systems, pp. 32-37, Mar. 2012.

[9] W. Chen, Y. Zhang, and Z. Xiong, ”Research and Realization ofthe Load Balancing Algorithm for Heterogeneous Cluster withDynamic Feedback,” Journal of Chongqing University, vol. 33,no. 2, pp. 2-14, 2010.

[10] S. Song, T. Lv, and X. Chen, ”A Static Load Balancing algorithmfor Future Internet,” Journal of Electrical Engineering,vol. 12, no. 6, 2014.

[11] D. Nurmi, R. Wolski, C. Grzegorczyk, G. Obertelli, S. Soman,L. Youseff, and D. Zagorodnov, ”The Eucalyptus Open-sourceCloud-computing System,” Proc. the 9th

IEEE/ACM InternationalSymposium on Cluster Computing and the Grid, pp.124-131, 2009.
[12] M. H. Willebeek-LeMair, and A. P. Reeves, ”Strategies forDynamic Load Balancing on Highly Parallel Computers,” IEEETrans. Parallel and Distributed, vol. 4, no. 9, pp. 979-993, Sept.1993.
[13] V. Shrivastava, P. Zerfos, K. W. Lee, H. Jamjoom, Y. H. Liu, andS. Banerjee, ”Application-aware Virtual Machine Migration inData Centers,” Proc. IEEE INFOCOM, pp. 66-70, 2011.
[14] M. Rahman, and P. Graham, ”Hybrid Resource Provisioningfor Clouds,” Journal of Physics: Conference Series, vol. 385, no.1, p. 012004, 2012.
[15] C. Dupont, G. Giuliani, F. Hermenier, T. Schulze, and A.Somov, ”An Energy Aware Framework for Virtual MachinePlacement in Cloud Federated Data Centres,” Proc. the 3th IEEEInternational Conference on Future Energy Systems: WhereEnergy, Computing and Communication Meet (e-Energy), pp.1-10, 2012.
[16] J. Zhao, Y. Ding, G. Xu, L. Hu, Y. Dong, and X. Fu, ”A LocationSelection Policy of Live Virtual Machine Migration for PowerSaving and Load Balancing,” The Scientific World Journal, vol.2013, no. 2013, p. 492615, Sept. 2013.
[17] CLOUDS. [Online]. http://www.cloudbus.org/cloudsim/