ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 5, May 2016

A Self-Adaptive Honey Bee Inspired Load-Balancing Algorithm in Cloud Data-Centre

Chandrakanta Korat[1], Daxa Vekariya[2]

PG Student, Dept. of CE, Noble Group of Institute, Junagadh, Gujrat, India[1]

Assistant Professor, Dept. of CE, Noble Group of Institute, Junagadh, Gujrat, India[2]

ABSTRACT: Load balancing is one of the keyissues in cloud computing. Scalability which is one of the very important features of cloud computing is enabled by load balancing Honey bee inspired load balancing algorithm is one of the dynamic approach. Bees Algorithm is a population-based search algorithm which uses the concept of swarm intelligence. The Honeybee algorithm is inspired by the "behaviour of a colony of honeybees foraging and harvesting food." This approach works well under heterogeneous types of resources but it does not show equivalent improvement in throughput while increasing number of resources. To overcome the above limitations, many researchers have put forward many improvement methods. Here proposed method uses the concept of Multi objective optimization for selecting the best solution with pre-emptive task scheduling which considers Time, Cost and Fault-rate for selecting optimal VM. For calculating the exact weight of each QoS parameter The Eigen Vector Method is used.This proposed system will be implemented in CloudSim as an virtual environment that will consider Multiple parameters of QoS value to choose the best virtual machine for task migration. It considers the task's priority while migrating the task from one virtual machine to another. Result shows that this approach reduces the overall-time and waiting time of task. So it increases the system-performance and reduces the makespan.

KEYWORDS: Cloud computing , Load balancing, Honey bee foraging behaviour, pre-emptive task scheduling

I. INTRODUCTION

Cloud computing brings benefits in "cost, flexibility and availability of service users" Those benefits makes the demand of Cloud services.[17] This demand raises many technical issues, like high availability and scalability. Load balancing is one of the keyissue in cloud environment. This mechanism distributes the dynamic local workload evenly across all the available nodes in the cloud. This will avoid the condition where some of the nodes are loaded heavily while others are idle or having little work. This helps to achieve a higher degree of resource utilization ratio and user satisfaction.[21] Hence, This will improve the resource utility and overall performance of the system. It also confirms that each computing resource is distributed fairly and efficiently. As a key concern in these issues, cloud load balancing allows cloud to "scale up to increasing loads" by efficiently assigning dynamic local workload fairly across all the available nodes. Scalability which is one of the most important characteristic of cloud computing, which is also enabled by load balancing. Hence, improving the performance and resource utility of a distributed system in such a way will decrease the carbon footprints and energy consumption to reach Green computing. Static load balancing algorithms generally don't consider the dynamic changes of these characteristics at the run-time. In addition, these static load balancing algorithms cannot adjust to load changes during the run-time. [2] Honey bee behavior inspired load balancing is one of the dynamic approach that improves the overall throughput of system and priority based load balancing attentions on falling the amount of time a task has to wait on a queue of the VM. Hence, it decreases the response of time of VMs

.

II. RELATED WORK

Load balancing improves the performance of the system by shifting the workload among the available processors. Machine's Workload means the total processing time it requires to execute all the tasks allocated to the node. Load balancing is done so that every node in the cloud does the same quantity of work thus

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 5, May 2016

increasing the throughput and minimizing the response time. Load balancing is one of the key factors to amplify the performance. Load balancing of virtual machines evenly that means any available machine is not partially loaded or lazy while others are heavily loaded. One important issue of cloud computing is to divide the workload at run-time. The advantage of workload distribution includes increased overall performance and resource utilization ratio and thereby attaining maximum client satisfaction.

Many researchers have put their attention on honey bee inspired load balancing techniques. Some of them are as follows.

1 In [4] authors presents a Priority based dynamic load balancing .It works better for heterogeneous cloud computing systems and is for balancing non-preemptive self-governing tasks . But considers just priority as the key QoS parameter

2 In [5] author used Random sampling method to choose VMs and iteration formulation to select better VM .It makes sure the system is under the balanced circumstance and finishing time is less.

3 In [1]. Pareto dominance principle is used.Not suitable for pre-emptive dependent tasks.

4. In [2] Pre-emptive Task Scheduling is used.The algorithm cannot work with dependent tasks and there is a QoS uncertainty.

5. In [7] Proposed algorithm uses the knowledge of previous solutions to look for better solutions. But it ignores the inactive condition of a Virtual machine even after the completion of jobs assigned to it

6. In [8] authors presents Random stealing method by which Idle time of Virtual machine is being saved.

7. In[6] authors presents a method which uses AntNet and Distributed Genetic Algorithm (DGA). It removess the requirement for having an entry for each destination node in the routing table.

8. In [3] proposed methods uses approach of submitting the tasks to the virtual machine till the machine gets overloaded. There is improvement in execution time and also reduction in waiting time of tasks

III. HONEYBEE'S BEHAVIOUR

The main steps of the Artificial Bee Colony algorithm are as below [22]:

- Initialize the Population
- REPEAT
  - Assign the employed bees on the food sources
  - Assign the onlooker bees on food sources depending on the quantity of nectar
  - Direct the scout bees to the site area for searching new food sources
  - Memorize the best or optimal food source found so far
- UNTIL requirements are met

In this algorithm, every cycle of the discovery contains this three steps, to send the employed bees to their food sources and identifying their nectar quantity and quality, after sharing the information of nectar of food sources, the finale selection of the best food source regions by the onlooker bees and evaluating the nectar amount of the food sources, defining the scout bees and then sending them randomly towards possible new food sources. At the first stage, a set of food sources is randomly selected by the bees and their nectar suitability are determined.

IV. PROPOSED SYSTEM

Honey bee inspired load balancing is a dynamic load balancing algorithm[4].But in traditional Honey bee inspired algorithm, there are limitations like uncertainty in quality parameters and also there is no improvement in the throughput of the system as expected. And most of them works only in non-preemptive manner. So, to overcome above problems proposed algorithm is developed which supports other QOS parameter while selecting optimal VM at IAAS level and can also work with preemptivetasks.The proposed algorithm works in **preemptive manner** and considers tasks priority while migrating them from one node to another.The proposed algorithm considers **multi**

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 5, May 2016

objective optimization for selecting optimal VM for load balancing and for assigning priorities to the task. Results can be evaluated in CloudSim[14].

- Multi-objective –optimization

  Multi-objective optimization is a part of multicriteria decision making problem, which involves more than one objective function that can be optimized at the same time. The multi-objective optimization problem can be stated as , min[f1(x),f2(x),...,fn(x)], x∈S,

  where the integer n is the total number of objectives and the set x is the feasible set of decision vectors. The feasible set is generally assigned by constraint functions.

- Pareto Dominance

  In multi-objective optimization, most of the times there does not exists a feasible solution which minimizes all of the objective functions at the same time. [1] Therefore, Pareto optimal solutions are used , Pareto dominance means a vector J1 = (j1, j2…jn) is said to dominate J2 = (k1, K2 ….kn) if and only if J1 is partially less than or equal to J2. That is J1i<=J2ifor all i ϵ{1, 2, 3…k}[1].There are many methods of finding pareto optimal solution and one of them is weighted sum method.

- Weighted Sum manner/Scalaraization method

  The weighted sum process to resolve the multi criteria optimization is as follows , [1]

  $$U = \sum_{i=1}^{k} WiFi(x)$$

  Here Wi is the weight and $\sum_{i=1}^{k} Wi = 1$ where Wi >0, i=1,...,n. Fi(x) are the objective function. For example, if k=3 then,

  $$U=w_1F_1(x) + w_1F_1(x) + w_1F_1(x),$$

  Here , w1 + w2 +w3 = 1.This technique is the simplest approach and possibly the most widely used traditional method. What should be the value of the weights ? It depend on the relative importance of each objective.[2]

- Preemptive and non preemptive task scheduling[3]

  To preempt means to act to prevent something or to replace something. On preemptive task scheduling characteristics are (1)Once in running state, task will continue to run.
  (2)Potential to monopolize the CPU (3)f higher priority task arrives than task will be put in queue and it has to wait. Preemptive task scheduling is exactly opposite with non preemptive task scheduling. Their characteristics are (1)Currently running process may be interrupted and put into ready state if higher priority task arrives (2) Preemption takes place only when priority of removed task is higher than running task.

The weighted sum method is used 1) To select optimal VM. 2) To calculate priority of a task

1) For selecting optimal VM

   $U=w_1T(i,j) + w_2C(i,j) + w_3F(j)$ -------(1)

   Here w_1,w_2 and w_3 presents the weight of the function.

   w_1,w_2,w_3> 0 and $w_1 + w_2 + w_3 = 1$

   So w_1,w_2,w_3ϵ [0,1]

w_1,w_2,w_3 shows the importance or weights of functions T(i,j), C(i,j) and F(i,,j) respectively
The Functions T(i,j),C(i,j) and F(i,j) denote efficiency ratios of time ,cost and fault respectively.

$$T(i,j) = \frac{PT(ti,mj)-tmin}{tmax-tmin} \qquad (1.1)$$

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 5, May 2016

$$C(i,j) = \frac{C(ti,mj)-Cmin}{Cmax-Cmin} \qquad (1.2)$$

$$F(j) = \frac{F(mj)-Fmin}{Fmax-Fmin} \qquad (1.3)$$

- Where t_{min} and t_{max} denotes the minimum and maximum execution time respectively. Similarly, C_{max} and C_{min}denotes maximum and minimum cost of the task. And likewise F_{min} and F_{max} are the minimum and maximum fault rate among all VM. And Assign task t_i to VM, m_j which has less minimization function value.

- Where $PT(t_i,m_j)$ is the required processing time of task ti on VM mj. It strictly depends on its own processing time and the processing time of higher priority task than t_i .$C(t_i,m_j)$ denotes the cost of executing task t_i on VM m_j[1].

- $C(t_i,m_j)= ɞ * PT(t_i,m_j) * V_{co} * Cm_j/Cclcap$ ……..(1.2.1)

- Clcap represents the capacity of the VM having less capacity and V_{co} is the cost of that VM which is calculated through fixed price allocation scheme. The Symbolɞ is the random variable.Most of the cloud providers assign virtual machine instances to the users by applying technique of fixed-price allocation schemes. It means that users has to pay flat prices per unit time for deploying that instances. They categorize different types of VM instances by their memory range ,number of processors, the bandwidth allotment, speed of processors, etc.[1]
- $F(m_j)$ is the fault rate.

2) For calculating priority of a task

We compute the priority of tasks by this equation:

$$P(i) = w_1* TU_i + w_2* TP_i + w_3* TL_i + w_4* LT_i \quad \text{…….(2)}$$

- Where P(i) stands for task T_i's priority

- w1,w2,w3,w4 Є[0,1] are weights of priority
- w1+w2+w3+w4=1.
- TUi, TPi, TLi, LTi are the task user type, task expected priority, task length and latency time of task Ti respectively.

EIGEN VECTOR METHOD

The most common and popular method to evaluate a priority vector is that proposed by Saaty, according to which the priority vector must be the principal eigenvector of A. Taking a matrix **A** whose entries are accurately determined as ratios between weights and multiplying it by **w** so we get

$$Aw= \begin{pmatrix} w_1/w_1 & w_1/w_2 \ldots w_1/w_n \\ w_2/w_1 & w_2/w_2 \ldots w_2/w_n \\ \cdots & \\ w_n/w_1 & w_n/w_2 \ldots w_n/w_n \end{pmatrix} \begin{pmatrix} w_1 \\ w_2 \\ \cdot \\ w_n \end{pmatrix} = \begin{pmatrix} nw_1 \\ nw_2 \\ \cdot \\ nw_n \end{pmatrix} = nw$$

Consequently to find the estimate there is a necessity to resolve general eigenvector equation: $Aw= \lambda_{max} w$

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 5, May 2016

where $\lambda_{max}$is the principal eigenvalue. Now, for the arbitrary positive reciprocal matrix A the value $\lambda_{max}$is always real, unique and not smaller than n.If the entries of A are ratios between weights, then the weight vector is the eigenvector of A related with the Eigen value n. What is the relative importance? They are asked to choose whether cost or any objective is very strong important, more important, as important, and so on down to very much less important. Each of these decisions is assigned a number on a scale. One common scale is shown in below Table 1. This is generally known as 1 to 9 scales.

For example, here for choosing the optimal VM, There is a minimization function which has 3 QoS parameter Cost, Time and Fault rate and w_1,w_2,w_3 shows its weights or importance respectively. So, for getting w_1,w_2 and w_3 mainly 3 comparison is needed for matrix comparisons:

1. Time to Cost
2. Time to Fault rate
3. Cost to fault-rate

Suppose time has strong importance than cost, so Time to cost comparison is 5

Table: 4.5 1 to 9 scale

Intensity of Importance	Definition
1	Equal importance
3	Moderate important
5	Strong important
7	Very Strong important
9	Extreme mportant
2,4,6,8	Intermediate values
Reciprocal of above	- if i is 3 compared to j - then j is 1/3 compared

LOAD BALANCUNG ALGORITHM

- Step1: Find the load and capacity of all VMs. Then check the value of 'σ' and define if the system is balanced or not. If balanced then exit.
- Step2: Take load balancing decision based on load. If load>max.capacity then exit.
- Step3: Group VMs based on loads.
- Step4:Apply Load balancing
  - Find out demand of each VM in OVM
  - Sort VMs in OVM
  - Sort VMs in UVM
  - If more than one VMs are in UVM
    - **VMd=Call Pareto optimal VM finding -A**
    - **Preemptive scheduling of the tasks- ---B**
- Step5:Update the no. of tasks assigned to VM_d
- Step6:Update sets OVM,BVM and UVM

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 5, May 2016

FIG 4.6 shows the flow of the algorithm step by step. First it checks the system's overall load to check that whether the system is balanced or not. If it is not balanced than the load balancing will be carried out.

- Step 1 in detail

Capacity of a VM, m is[1],

$$C_i=P_{ni}\times P_{mpi}+VM_{bwi} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(A)$$

Where ,P_{ni} is the number of processors in VM_i,P_{mpi} is millions of instructions per second of all processors in VM_i,VM_{bwi} is the communication bandwidth ability of VM_i Capacity of all VMs,

$$C = C_1+C_2+.....C_n \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots..(B)$$

Load on a VM m is, Total length of tasks that are assigned to a VM is called load, [1] The processor load is given by the number of tasks simultaneously running on that processor.[20]Load of a VM at time t can be calculated as the Number of tasks at time t on service queue of VM_i divided by the service rate of VM_i at time t.

$$Lvmi,t = \frac{N(T,t)}{S(vmi,t)}. \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots..(C)$$

Loads of all Vms,

$$L = Lvm_1+Lvm_2+.....Lvm_n \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots(D)$$

Processing time of a VM m is,

$$PT_i = Lvm_i/c_i \quad \ldots\ldots.,\ldots\ldots\ldots\ldots\ldots\ldots..(E)$$

Processing time of all VMs:

$$PT=L/C \quad \ldots\ldots \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(F)$$

Standard deviation of load: $\sqrt{1/m\sum_{i=1}^{m}(PTi-PT))^2}$ ……(3)

If the standard deviation of the load of VM σ is under or equal to the threshold condition, whose value is in [0–1] then the system is balanced. Otherwise system is in an imbalance state, overloaded or under loaded.

When the current workload of VM group beats the maximum capacity of group which is predefined by threshold value, then the group is overloaded. Load balancing is not possible in this case. This condition is checked in **Step 2.**

- **Sub Algorithm A**- Pareto optimal VM Finding

Step1: Identify the cost of all VMs using Fixed Price Allocation Scheme.
Step2: For each VM m_j, For each task t_i,
Calculate execution cost of t_i on VM m_j using (1.2.1).
Calculate the minimization function U
Select m_j having minimum U
Step3: Return m_j.

- **Sub Algorithm B**- Preemptive task scheduling

For each task T in VMs , search machine VM_dC UVM.
If (Incoming task priority > =priority of task running on VM
If (Remaining expected completion time of incoming task <Task currently running on VM)
-Save the state of currently executing task and
-Preempt it
-Assign incoming task to VM and execute it

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 5, May 2016

Return

- Load Balancung Algorithm
  - Step1: Find capacity and load of all VMs. Then check the value of 'σ ' and determine whether the system is balanced or not. If balanced then exit.
  - Step2: Take load balancing decision based on load. If load>max.capacity then exit.
  - Step3: Group VMs based on loads.
  - Step4:Apply Load balancing
    - Find demand of each VM in OVM
    - Sort VMs in OVM
    - Sort VMs in UVM
    - If there are more than one VM in UVM
      - **VMd=Call Pareto optimal VM finding ------ A**
      - **Preemptive scheduling of the tasks- --------B**
  - Step5:Update the no. of tasks assigned to VMd
  - Step6:Update sets OVM,BVM and LVM

    FIG 01 shows the flow of the algorithm syep by step.Firstiy checks the systems overall load to check that whether the system iis balanced or not.If it is not balanced than the load balancing will be carried out.

  - Step 1 in detail

    Capacity of a VM, m is[1], Ci=Pni×Pmpi+VMbwi

    Where ,Pni is the number of processors in VMi,Pmpi is millions of instructions per second of all processors in VMiVmbwi is the communication bandwidth ability of VMi Capacity of all VMs, C = C1+C2+.....Cn

    Load on a VM m is, Total length of tasks that are assigned to a VM is called load, [1] The processor load is given by the number of tasks simultaneously running on that processor.[20]Load of a VM at time t can be calculated as the Number of tasks at time t on service queue of VMi divided by the service rate of Vmi at time t. $Lvmi,t = \frac{N(T,t)}{S(vmi,t)}$. Loads of all Vms, L = Lvm1+Lvm2+.....Lvmn

    Processing time of a VM m is, PTi =Lvmi/ci ,Processing time of all VMs:PT=L/C

    Standard deviation of load: $\sqrt{1/m\sum_{i=1}^{m}(PTi - PT))^2}$ (3)

    If the standard deviation of the VM load σ is under or equal to the threshold condition , whose value is in [0–1] then the system is balanced. Otherwise system is in an imbalance state, overloaded or under loaded

  When the current workload of VM group exceeds the maximum capacity of the group which is predefined by threshold value, then the group is overloaded. Load balancing is not possible in this case. This condition is checked in **Step 2.**
- Sub Algorithm A- Pareto optimal VM Finding

  Step1: Find cost of all VMs using Fixed Price Allocation Scheme.
  Step2: For each task ti, For each VM mj,
  Calculate cost of executing ti on mj using (1.2.1).
  Calculate the minimization function F
  Select mj having minimum F
  Step3: Return mj.
- Sub Algorithm B- Preemptive task scheduling

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 5, May 2016

For each task T in VMs find machine VMd Є UVM such as T is preemptive.
If (Incoming task priority > =priority of task running on VM
If (Remaining expected completion time of incoming task <Task currently running on VM)
-Save the state of currently running task and
-Preempt it
-Allocate incoming task to VM and execute it
Return

V. RESULT AND DISCUSSION

The proposed algorithm is implemented using cloudSim simulator which runs on NetBeans IDE 7.2.1.CloudSim is a latest, widespread, and extensible simulation framework that allows faultless modeling, simulation, and testing of emerging Cloud computing infrastructures . By using CloudSim, researchers can check the performance of a newly developed application service in a restricted and easy to set-up surroundings. Based on the evaluation results reported by CloudSim, they can then advance fine tune the service performance .It simulation of large-scale Cloud computing environments, including data centers, on a single physical computing node.It allows to tune the performance bottlenecks before real-world deployment on commercial clouds.The proposed algorithm works better with multiple qos parameters. Here we consider fault rate of VM. So the system become more fault tolerant by using this algorithm.Figure Shows the that proposed algorithm shows the fair distribution of load.

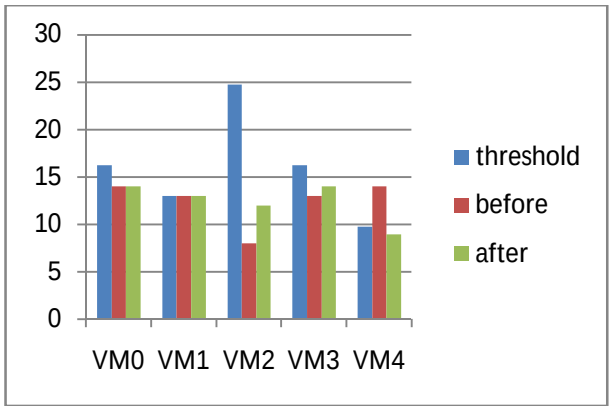

Fig 1. Load Distribution

VI. CONCLUSION AND FUTURE SCOPE

In Cloud computing unbalanced load results in degradation of the system performance. Performance of the system can be improved by bringing the overloaded VMs to balanced state and effectively utilizing the underloaded and idle VMs. Here the proposed algorithm follows the foraging behaviour of honey bees for allocating VMs to the tasks and uses pre-emptive task scheduling. Pareto dominance concept is used for both selecting optimal VM and for setting priorities to the tasks. Here, the proposed algorithm uses the Eigen Vector method to calculate the weight of each QoS value. The priority of the independent tasks is considered while pre-empting the tasks to achieve minimum makespan and maximum throughput. When pre-empting tasks remaining expected completion time of the tasks and priority is considered so that the waiting time of the tasks can be reduced this will also lead to improve the performance of the datacenters.

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 5, May 2016

REFERENCES

[1] Sheeja Y S, Jayalekshmi S Cost Effective Load Balancing Based on honey bee Behaviour in Cloud Environment. IEEE 2014 First International Conference on Computational Systems and Communications (ICCSC) | 17-18 December 2014 | Trivandrum
[2] G.Shobana, M.Geetha, Dr.R.C.SugantheNature Inspired Preemptive Task Scheduling for Load Balancing in Cloud Datacenter . IEEE ICICES2014 - S.A.Engineering College, Chennai, Tamil Nadu, India
[3] Harshit Gupta, KalicharanSahu,Honey Bee Behavior Based Load Balancing of Tasks in Cloud Computing International Journal of Science and Research (IJSR) Volume 3 Issue 6, June 2014
[4] L.D. DhineshBabu, P. VenkataKrishna,Honey bee behavior inspired load balancing of tasks in cloud computing environments, Appl. Soft Comput. J. (2013)
[5] Jeng-Shyang Pan, Haibin Wang, Hongnan Zhao, and Interaction Artificial Bee Colony Based Load Balance Method in Cloud Computing, Springer Advances in Intelligent Systems and Computing 329,2015
[6] Kamlesh Kumar Pathak, Prasant Singh Yadav, RameshwaramTiwari, Dr.Tarun Kr. GuptaA Modified Approach for Load Balancing in Cloud Computing Using Extended Honey Bee Algorithm, IJRREST: International Journal of Research Review in Engineering Science and Technology (ISSN 2278- 6643) | Volume-1 Issue-3, December 2012
[7] Ms. Anna Baby,Improved Honey Bee inspired load balancing of tasks with position updation, International Journal for Research in Applied Science & Engineering Technology (IJRASET) Volume 3 Issue IV, April 2015
[8] Ms. Anna Baby, Dr. Joshua Samuel Raj,An efficient load balancing using Bee foraging technique with Random stealing , IOSR Journal of Computer Engineering (IOSR-JCE)e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 17, Issue 2, Ver. V (Mar – Apr. 2015), PP 97-104
[9] RajkumarBuyyaet. el., Cloud Computing: Principles and Paradigms, Wiley India Edition
[10] PriyadarashiniAdyashaPattanaik , Sharmistha Roy ,Performance Study of Some Dynamic Load Balancing Algorithms in Cloud Computing Environment, 2nd International Conference on Signal Processing and Integrated Networks (SPIN),2015
[11] Timothy Marler·Jasbir S. AroraThe weighted sum method for multi-objectiveoptimization: new insights, springer Struct Multidisc Optim (2010) 41:853–862
[12] Giuseppe Narzisi, Classic Methods for Multi-Objective OptimizationCourant Institute of Mathematical SciencesNew York University31 January 2008
[13] Operating Systems – Scheduling ECE 344 – Week
[14] Rodrigo N. Calheiros, Rajiv Ranjan, CloudSim: a toolkit for modeling and simulation of cloudcomputing environments and evaluation of resourcerovisioning algorithms, SOFTWARE – PRACTICE AND EXPERIENCESoftw. Pract. Exper.2011;41:23–50Published online 24 August 2010 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/spe.995