ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 8, August 2015

A Novel Honey Bee Inspired Algorithm for Dynamic Load Balancing In Cloud Environment

Chandrakanta Korat[1], Piyush Gohel[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: In recent years swarm intelligence (SI) becomes important for many researchers. Artificial bee colony algorithm (ABC) is one kind of swarm based algorithm. It follows the foraging behaviour of honey bees. In cloud computing environment, the load balancing is an important factor. The tasks are executed on Virtual Machines (VMs) and those VMs are run in parallel manner, so that the load across all VMs has to be balanced well. Because in cloud environment user pay for the service ,reduction in the execution time of task and the cost of using VM instances (CPU, Memory, Bandwidth etc) is most important.. The aim of proposed algorithm is to maximize the throughput. In this paper, we have discussed different honey bee inspired algorithms. Here the proposed algorithm uses the concept of pareto dominance's weighted sum approach for selecting optimal VM and also work with preemptive task scheduling.

KEYWORDS: Cloud computing , Load balancing, Honey bee foraging behavior.

I. INTRODUCTION

Cloud computing is an internet based distributed computing and uses pay-as-you-go model. Buyya have defined it as follows: "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. [9]" The main purpose behind this model is offering computing, storage, and software "as a service". The advantages of cloud computing are accessing huge range of applications without downloading or installing them, can access the application from anywhere in the world, avoid expenditure on purchasing new hardware or software and can share the resources.

Cloud Computing become so admired now a days. It provides a flexible way to retrieve and keep data. As a result, it is significant to do research in some areas of the cloud to get better the storage utilization and the performance for the users. One important issue related with this load balancing field is to apply dynamic load balancing.

II. LOAD BALANCING IN CLOUD COMPUTING

Load balancing is the strategy of dividing the workload between many computers or datacenters equally in order to enhance the performance and makes the process faster. Many algorithms are proposed through which load can be distributed equally and with minimum Response time.

Load balancing concept is classified into two class i.e. static load balancing algorithm and dynamic load balancing algorithm. Static algorithm checks the current state of the node and distributes the requests on a fixed set of rules depending on the input requests. Second type is the dynamic Load Balancing algorithm which is also known as self-adaptive algorithm. This algorithm checks the previous state and also the current node and adjusts traffic distribution evenly in real time. Honeybee inspired algorithm is one of the dynamic technique.[4]

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 8, August 2015

III. HONEYBEE'S BEHAVIOUR

The artificial bee colony algorithm(ABC), is an optimization algorithm which is based on the smart foraging behavior. This Meta heuristic is inspired by the smart foraging behavior of honey bee swarm. The ABC contains three groups of bees [1]. They are

- Scout bees:-Simple meaning of scout is a soldier sent to gather information. The bee who carries out random search is identified as scout. Upon finding one they returned to bee hive and inform forager bees.
- Forager bees: - The bee which is going to the food source which is visited by scout bees previously is forager bee.
- Onlooker bees: - Onlooker means somebody who watches an event without participating in it**.** The bee waiting on the dancing area is an onlooker bee.

Scout bees are sent to search for suitable sources of food, when one is found, they return to the hive to advertise this using a display dance known as a "waggle dance". This dance shows the appropriateness of the food source. Forager bees then trail the scout bees back to the discovered food source and begin to collect it .The remaining amount of food available is reflected in their waggle dances, on the bases of how many bees have been returned to the hive which allows more bees to be sent to a source, or exploited sources to be discarded.

IV. LITERATURE SURVEY OF HONEYBEE INSPIRED LOAD BALANCING ALGORITHMS.

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

V. 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 preemptive tasks. The proposed algorithm works in **preemptive manner** and considers tasks priority while migrating them from one node to another. The proposed algorithm considers **multi 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 , $\min[f1(x),f2(x),...,fn(x)]$, $x \in S$,
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. [18] 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 , [11]

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

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,

$$U=w1F1(x) + w1F1(x) + w1F1(x),$$

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.[12]

- Preemptive and non preemptive task scheduling[13]
  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.

- Applying weighted sum method
  Here weighted sum method is applied 1)For selecting optimal VM and 2)For calculating priority of a task.
  1) For selecting optimal VM

$$U=w1T(i,j) + w2C(i,j) + w3F(j) \qquad (1)$$

Where ,

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

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

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

  - Where Where w1,w2,and w3 are related weights which shows the importance of that objective functions T(i,j),C(i,j) and F(i,j) respectively. T (i,j), C (i,j) and F(i,j) represent cost-efficiency ratios of time ,costs and fault rate respectively. tmin and tmax are the minimum and maximum execution time respectively Cmax and Cmin are maximum and minimum cost of any task. And Fmin and Fmax are the minimum and maximum fault rate among all VM. And Assign task, tVM, mj which has less minimization function value.

  - Where PT(ti,mj) is the required processing time of task ti on VM mj. It depends on the processing time of higher priority task than ti and its own processing time (ti,mj) represents the cost of executing task ti on Vm mj [1].

  - C (ti,mj)= ϭ * PT(ti,mj) * Vco * Cmj/Cclcap (1.2.1)

Clcap denotes the capacity for the VM having less capacity and Vco is the cost of that VM which is calculated through fixed price allocation mechanism. σ is the random variable. Most of the cloud providers assign virtual machine instances to their users by using technique of fixed-price allocation schemes. It means that users has to pay flat prices per unit of time for using that assets. They classify different types of VM instances by their number of processors, speed of processors, the memory range, the bandwidth allotment, etc.[1]

- FR(mj) is the fault rate. FR(mj) = n(f)/t. It gives the number of faults occurs in VM mj in given period of time

2) For calculating priority of a task
We compute the priority of tasks by this equation:

$$P(i) = w1 * TUi + w2 * TPi + w3 * TLi + w4 * LTi \quad (2)$$

- Where P(i) stands for task Ti's priority
- w1,w2,w3,w4 €[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.

FIG 01 shows the flow of the algorithm syep by step.First iy 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.

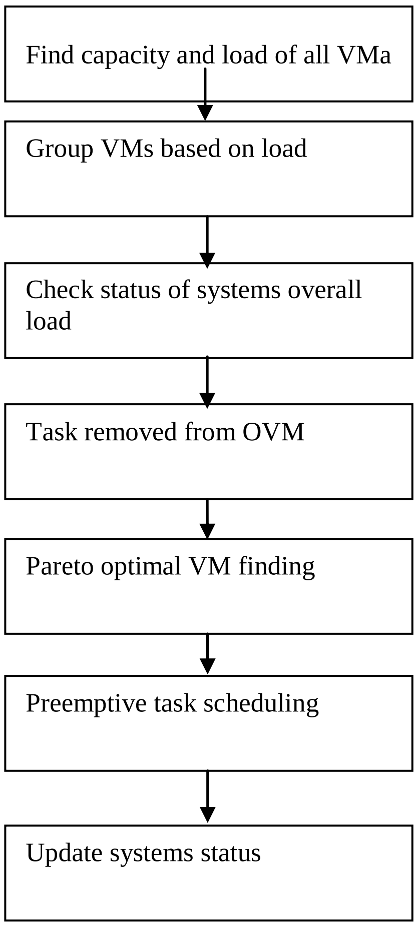
Find capacity and load of all VMa
Group VMs based on load
Check status of systems overall load
Task removed from OVM
Pareto optimal VM finding
Preemptive task scheduling
Update systems status

FIG 01

- Load Balancung Algorithm
  - Step1: Find capacity and load of all VMs. Then check the value of 'ϭ ' 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.First iy 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 VMi Vmbwi 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 ϭ 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

```
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
```

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 8, August 2015

```
                    -Preempt it
                    -Allocate incoming task to VM and execute it
Return
```

VI. 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. Here VMs are considered with processing power of 1000-7000MIPS. Weights are considered as 0.5, 0.3, and 0.2 for w1, w2 and w3 repectively.

VII. CONCLUSION AND FUTURE SCOPE

Here the proposed algorithm follows the foraging behaviour of honey bees for allocating VMs to the tasks and uses preemptive task scheduling. Pareto dominance concept is used for both selecting optimal VM and for setting priorities to the tasks and here multiple QOS parameters are considered. The priority of the independent tasks is considered while preempting the tasks to achieve minimum makespan and maximum throughput. When preempting 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.

Here tasks are considers as independent tasks. In future work this honey bee inspired load balancing algorithm can be extended for dependent tasks .Calculation of assigning priority of task and finding optimal VM can be improved by considering other QOS parameters. There are many others method for getting pareto optimal solution. Algorithm can be develped for those methods

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, Kalicharan Sahu,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. Dhinesh Babu, P. Venkata Krishna,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, Rameshwaram Tiwari, 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] Rajkumar Buyya et. el., Cloud Computing: Principles and Paradigms, Wiley India Edition

[10] Priyadarashini Adyasha Pattanaik , 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