ISSN:0975-9646

Monika Rathore et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 6 (4) , 2015, 4128-4132

Load Balancing of Virtual Machine Using Honey Bee Galvanizing Algorithm in Cloud

Monika Rathore, Sarvesh Rai, Navdeep Saluja

Computer Science and Engineering, Infinity Management & Engineering College

Abstract -Load balancing is most important task of cloud computing. In order to attain best machine utilization, tasks from overloaded virtual machines ought to be transferred to under loaded virtual machines. Scheduling of resources are very massive problem on cloud. Scheduling of the models, cost, quality of service, time, and conditions of the request for access to services are factors is to be focused for cloud. This paper, the honey bee forage mechanism for load balancing is to improved load balancing in cloud to utilize its resources on cloud, is applied to optimize the scheduling of Virtual Machine (VM) on Cloud. The most focus is to research the distinction of Virtual Machine load scheduling to cut back the makespan of processing time that is total length of the schedule. Virtual Machine(VM) load is calculated and checked for confinement at intervals a where the threshold condition set. With honey bee forage methodology, tasks are purloined from a random Virtual machine once a VM is idle. This saves the idle time of the process parts within the Virtual machine. The scheduling strategy was simulated using CloudSim tools. Experimental results indicated that the mixture of the planned using honey bee forage behavior and scheduling supported the dimensions of tasks performed an scheduling strategy in ever changing atmosphere and leveling work load which may reduce the span of processing time.

Keys—**Artificial Bee Colony, Cloud Computing, programing Management, Virtualization Machine**

I. INTRODUCTION

Cloud computing is one in all distributed computing paradigm that primarily focuses on providing everything as a service to the client and it provides computational and storage resources to users. The processing machines in cloud environments are referred to as virtual machines (VMs). Scheduling of the client tasks to offered resources may be a difficult task. Many tasks are allotted to one or a clusters of VMs that run the tasks at the same time by Virtual Machine [1]. This type of environments ought to make certain that the tasks are well balanced in virtual machines. Load balancing is that the task of distribution of application tasks to totally different processors to reduce program execution time. Effective implementation of load balancing will build cloud computing more practical and it conjointly improves user satisfaction. Load balancing distributes workloads across multiple computing resources like computers, a laptop cluster, network links, central process units or disk drives. Load balancing aims to optimize the resource use, maximize the makespan further on minimize the latency. The planned algorithm improves the honey bee forage technique by guaranteeing that no virtual machines stay idle. The performance of scheduling and allocation policy on a cloud infrastructure is a particularly difficult problem to tackle.

CloudSim: Simulation structure that allows perfect modeling and simulation of Cloud computing infrastructures and running services. The simulation structure has the subsequent novel features:

- Support for modeling and illustration of enormous scale Cloud computing infrastructure
- Complete platform for modeling knowledge centers, scheduling, and allocations.
- Accessibility of virtualization engine
- Flexibility to modify between space-shared and time-shared allocation.

II. RELATED WORKS

Load balancing mechanism distributes the work across multiple computing resources to utilize them effectively and to cut back the latency of the task, at the same time eliminating a condition during which bound nodes are over loaded whereas others are beneath loaded.

Dhinesh man and P.Venkata Krishna [1] Algorithm for Honey Bee galvanized load balancing(HBB-LB) of tasks in cloud computing environments that aims to attain well balanced load across Virtual machines for increasing the outturn. In Particle Swarm improvement (PSO) planned by Ayed salman, Imtiaz Ahmed [5] combines native search ways with world search ways (through neighborhood experience), making an attempt to balance exploration and exploitation. The formula is planned for the task assignment drawback for solid distributed computing systems. The result shows that this runs quicker with less quality. Belabbas Yabougi and Meriem Meddeber [6] planned a distributed load balancing model for grid computing which may represent any grid topology into forest structure. Distributed approach gains are forever higher than those achieved by the hierarchic approach. Erik Cuevas, Daniel Zaldívar, Marco Pérez-Cisneros, Humberto Sossa, Valentín Osuna [7] introduces a Block matching Algorithm for motion estimation supported Artificial Bee Colony (ABC). Here the numbers of search locations are drastically reduced by considering a fitness calculation. This Algorithm indicates once it's possible to calculate or solely estimate new search locations. It reduces the procedure quality. Guopu.Zha, surface-to-air missile Kwong [8] planned Gbest guided artificial bee Colony Algorithm for numerical perform improvement incorporating the knowledge of worldwide best (gbest) answer into the answer search equation to enhance the exploitation. GABC Algorithm consists of the 3 totally different stages that are the utilized bee stage, spectator stage and therefore the scout stage. The spectator stage tends to pick out the nice answer to any update, whereas each the utilized bee stage and update each individual within the population. Pei-Wei Tsai, Jeng-Shyang Pan, Bin-Yih Liao dynasty, Nursingd Shu-Chuan Chu [9] introduce an increased Artificial Bee Colony improvement. The spectator bee is meant to maneuver straightly to the picked co- ordinate indicated by the utilized bees and judge the fitness worth close to it within the original rudiment formula so as to cut back procedure quality. M.V Panduranga Rao, S.Basavaraj Patil [10] planned a dynamic tree primarily based model in load balancing methods for Grid computing

III. PROBLEM STATEMENT

Until recently the main works on load balancing assumed solid nodes. Many instances of Cloud computing, as outlined herein, wherever dynamic and heterogeneous systems are necessary to produce ondemand resources or services. The Amazon EC2, dynamic load balancing is handled by replicating instances of the precise middleware platform for internet services. This is often achieved through a traffic analyzer that tracks the time taken to method a shopper request. New instances of the platform are started once the load will increase on the far side predefined thresholds [2]. Therefore, combos of rules impose the circumstances and answer for load balancing. Because the systems increase in size and quality, these rule sets become unwieldy and it should not be potential to take care of a viable observation and response cycle to manage the procedure work. In short, the dimensions of those systems could exceed the capabilities of connected meta systems to take care of a sufficiently agile and with efficiency organized load balancing (or general management) rule set. Once such a lot of management rules are outlined at intervals a system, there are probably to be conflicts amongst the rules; interactions and impact are normally terribly tough to research. For example, the execution of one rule could cause an incident, triggering another rule or set of rules, passionate about current state. These rules could successively trigger any rules there's a possible for an infinite cascade of policy execution to occur. In addition these rules are static in nature; there's typically no provision for rule refinement or analysis. A system rule requiring alteration or adjustment necessitates the system or part being taken offline, reprogrammed and deployed into the system. A load balancing system is needed that self regulates the load at intervals the Cloud's entities while not essentially having to possess full information of the system. Such self organized regulation could also be delivered through distributed algorithms; directly enforced from naturally discovered behavior, specifically designed to take care of a globally-balanced load, or directly fixing the topology of the system to boost the natural pattern of load distribution.

IV. HONEY BEE BEHAVIOUR IN LOAD BALANCING OF TASKS

Effective implementation of load balancing will build cloud computing more practical and it conjointly improves user satisfaction. Within the planned methodology, a honey bee forage technique is employed for task allocation and cargo balancing. Once tasks are allotted to the VMs, current load is calculated. If the VM becomes overloaded the task is transferred to the neighborhood VM whose load worth is below threshold [14]. Honey Bee forage technique employs suburbanized load leveling methodology and task transfer are disbursed on the fly. The algorithm ensures performance of the system and avoid system imbalance.

A) BEE FORAGE BEHAVIOUR

The artificial bee colony formula (ABC) algorithm supported the intelligent forage behavior of honey bee swarm and was planned by Karaboga in 2005 [15]. The formula is totally galvanized by natural forage behavior of honey bees.

B) INITIALIZATION METHOD

Artificial Bee Colony algorithm starts by correlating all the bees with willy-nilly created food sources. bound food sources are indiscriminately elect by bees and their nectar quantity is set. These bees return onto the hive and share the knowledge with bees waiting in dance space [16]. Initialize the population of the scout bees, generate indiscriminately scout bees into the food sources and calculate the fitness values.

C) Algorithm

Repeat:

- every the utilized bees search round the food ources and update the new fitness, if the new fitness is best than the previous values.
- choose utilized bees and recruit on looks bees to go looking round the food sources and calculate on their fitness worth.
- select the onllkes bees with have the most effective fitness worth.
- Send scout bees into the food sources to get new food sources.
- Until (Stopping criterion isn't met)
  End At the start, the initial n scout bees are placed indiscriminately in VM on Cloud computing and n is that the range of scout bees.

E) EMPLOYED BEE SECTION

Employed bees be the food supply and supply the neighborhood of the supply in its memory. when sharing the knowledge within the dance space, utilized bees attend food supply visited by its previous cycle and select new food supply by victimization the knowledge within the neighborhood. Then spectator prefers a food supply counting on nectar info provided by utilized bees.

F) ONLOOKER BEE SECTION

Onlooker bees get the knowledge concerning food sources from utilized bees in hive and choose one in all the sources. spectator bee is anticipating a dance to decide on a food supply. Waggle/tremble/Vibration dances ar performed by the bees to relinquish a concept concerning quality and amount of food and its distance from bee hive.

G) SCOUT BEE SECTION

Scout bee disbursed random search. once the nectar supply is abandoned by the bees, a brand new food supply is indiscriminately determined by a scout bee.

V. ENHANCED HONEY BEE INSPIRED LOAD BALANCING ALGORITHM

Within the planned methodology, increased honey bee forage technique with random stealing is employed for task allocation and cargo leveling. Once tasks are allotted to the VMs, current load is calculated. If the VM becomes overloaded the task is transferred to the neighborhood Virtual machine whose load worth is below threshold [14]. Honey Bee forage technique employs suburbanized load balancing methodology and task transfer are disbursed on the fly. The formula so ensures performance of the system and avoid system imbalance.

Additionally, rudiment consists of 3 management parameters:

1) Population size (SN) is that the range of food sources (or solutions) within the population. metal is capable the quantity of utilized bees or spectator bees.

2)Most Cycle range (MCN) refers to the most range of generations.

3) Limit is employed to diversify the search, to see the quantity of allowable generations that every non-improved food supply is to be abandoned.

LetVM={VM1,VM2,VM3,…,VMN} is a set of N virtual machines and Task= {task1,2, task3, … ,K} of K task to be regular and processed in VM. All the machines ar unrelated however are paralleled

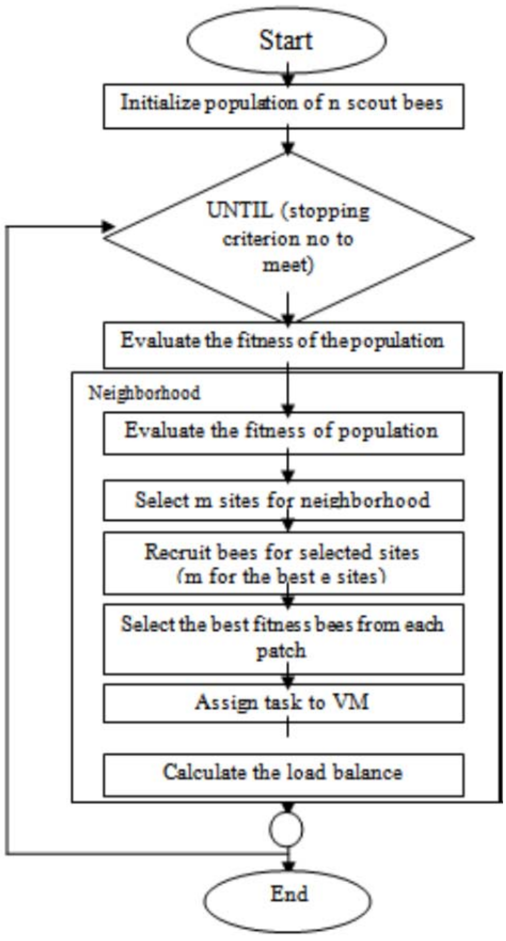

Fig 1. Flow sheet of VM algorithm and cargo balancing victimization rudiment.

VI. EVALUATE THE FITNESS OF THE POPULATION:

$$fit_{i,j} = \frac{\sum_{i=1}^{n} tasklength\ i,j}{\text{Evaluate capacity of VM}j(\text{capacity}j)} \quad \text{--- (1)}$$

Where $fit_{i,j}$ is that the fitness of the bees population of i in VM_j. tasklength is that the length of the task that has been submitted in VMj and capability is that the capability of VM_j calculating supported the subsequent

$$capacityj = pe\_numj \times pe\_mipsj + vm\_bwj \quad \text{---- (2)}$$

Select m Sites for Neighborhood Search: Scout bees with the best fitness are chosen as choose Bee" and therefore the sites visiting by them are chosen from neighborhood of m VMs. Recruit Bees for elect Site: Send a lot of bees to neighborhood of the most effective VM, then judge the fitness supported

$$fit_{i,j} = \frac{\sum_{i=1}^{n} tasklength\ i,j + inputfilelength}{\text{Evaluate capacity of VM}j(\text{capacity}j)} \quad \text{----(3)}$$

Where inputfilelength is that the length of the task before execution. choose the most effective Fitness Bees from every Patch and Assign Task to Virtual Machine: for every spherical, the bee with the best fitness are chosen to assign task in Virtual Machine.

Calculate Load Balance: when submitting tasks to the beneath loaded VMj, the present work of all offered VMs will be calculated by victimization the knowledge that received from the datacenter. Thus, variance (SD) is calculated so as to measure the deviations of load on VMs. variance of the load will be measured as

$$SD = \sqrt{\frac{1}{n}\sum_{j=0}^{n}(Xj - \bar{X})2} \quad \text{----- (4)}$$

Processing time of VM:

$$Xj = \frac{\sum_{i=1}^{k} \text{tasklength i}}{\text{capacityi}} \quad \text{----- (5)}$$

Mean of all processing times of all VMs:

$$X = \frac{\sum_{j=1}^{n} \text{Xj}}{\text{n}} \quad \text{----- (6)}$$

If the S.D. of the loaded VM is a smaller amount than or capable the mean, then the system is during a balance state. On the opposite hand, if the S.D. is above the mean, then the system is in imbalance state. The preventative programming is prioritized. The best priority method should be the method that's presently used. The preventative programming feature permits a unfinished high-priority job to preempt a running job of lower priority.

VII. EXPERIMENTAL RESULTS

According to the formula delineated higher than, the simulation victimization CloudSim-3.0.1 Tools are self-addressed. There are four servers are used here. The parameter setting of rudiment formula is as follows. The experiment is shown within the graph that consists of comparison of resource usage to the servers. The bar diagram shows the entire memory and therefore the used memory.

The of Makespan for load leveling victimization honey bee galvanized load balancing formula (HBBLB) is illustrated in Fig.2 Makespan will be outlined because the overall task completion time. we have a tendency to denote completion time of task T_i on VM_j as CT_{ij}.

$$\text{Makespan} = \max\{CT_{ij} \mid i \in Ti, j, i = 1,2,\ldots\ldots n \text{ and } j \in 1,2,\ldots\ldots m\}$$

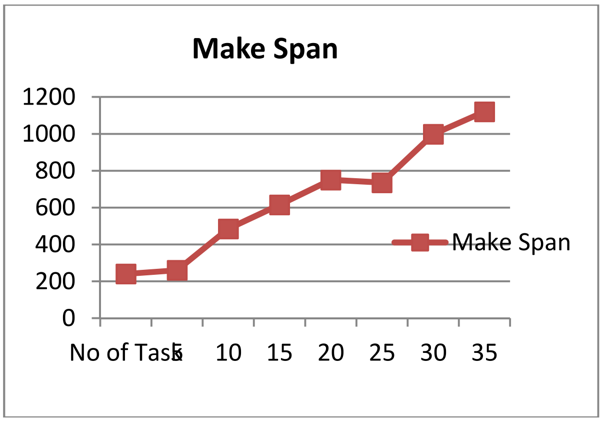

Fig 2 Makespan for Load Balancing using Honey Bee Algorithm

The coordinate axis represents the quantity of tasks and coordinate axis represents Makespan in milliseconds. With load leveling victimization honey bee galvanized load leveling (HBBLB), the makespan is reduced significantly. once range of tasks will increase, the distinction in makespan is a lot of, the latency in milliseconds for HBBLB . The coordinate axis represents range of tasks and coordinate axis represents the latency in milliseconds. it's the quantity of your time taken between submission of asking and therefore the initial response that's created. The reduction in waiting time is useful in raising the responsiveness of the VMs.

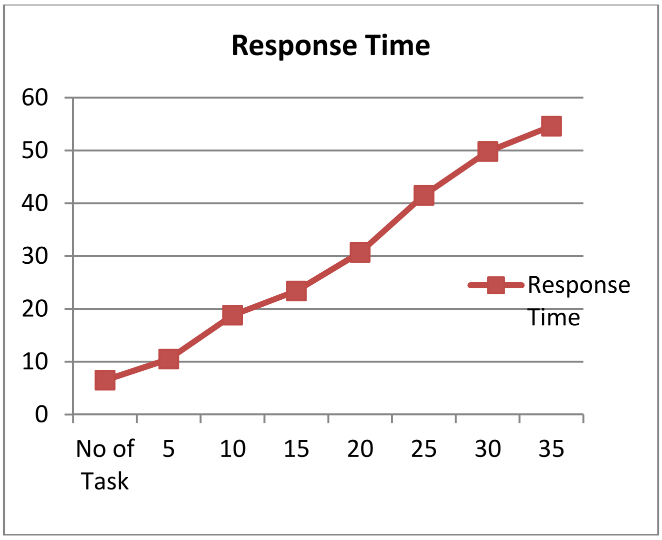

Fig 3 Response Time

The degree of imbalance in load in terms of range of tasks. The coordinate axis represents the quantity of tasks and coordinate axis represents the imbalance degree. Imbalance degree is outlined in equation (4)

$$\text{Degree of imbalance} = (T_{high} - T_{low})/T_{avg}$$

Where T_{high} is the best task, T_{low} is that the lowest task among all the virtual machines and T_{avg} is that the average task of virtual machines. From Fig.4

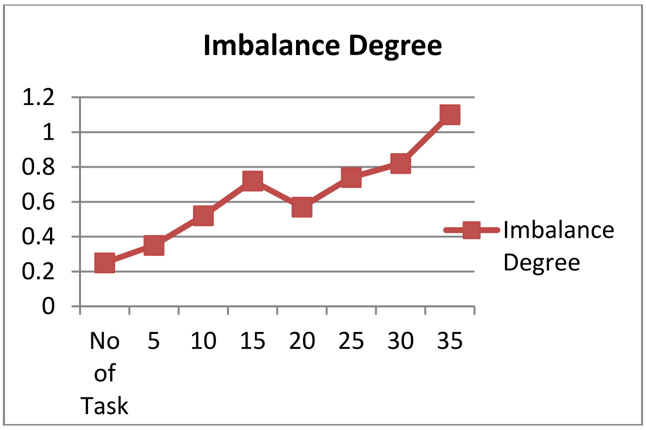

Fig 4 Degree of Imbalance

An important parameter utilized in this work to research the load leveling strategy of the planned formula is that the average resource utilization and is expressed in proportion.

Resource Utilization =VM demand / range of tasks

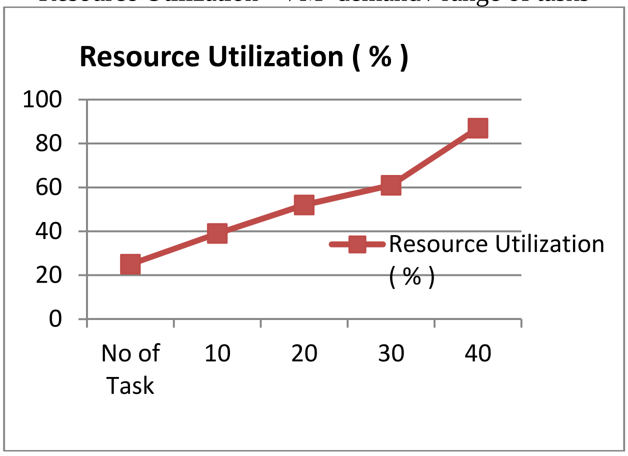

Fig 5 Resource Utilization

Fig.5 shows the resource utilization rate of honey bee load leveling methodology. Idle time is that the time between the days at that task is arrived on a virtual machine and time of task to be allotted to 1 bound virtual machine. The comparison is formed in terms of range of tasks and therefore the idle time and therefore the results are shown in Fig. 6.

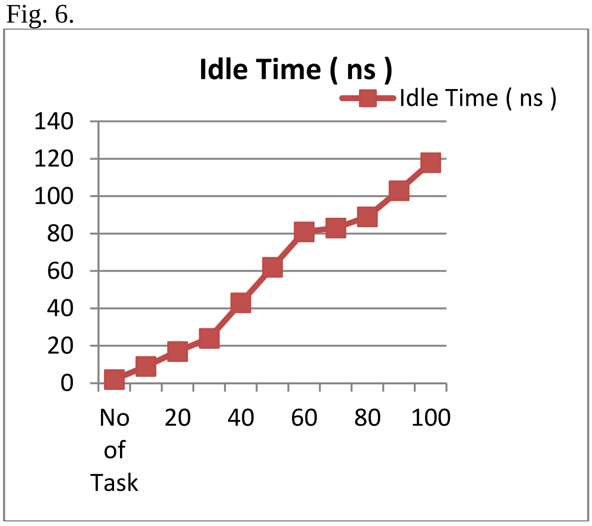

Fig. 6 Idle Time (ns)

VIII. CONCLUSION

This paper presents implementation formula which can resolve the Virtual machine programming management at a lower place the dynamic atmosphere of the amount of VMs and requests on Cloud computing. We have projected a flow chart for load balancing in cloud computing environments supported behavior of honey bee forage strategy. The tasks are to be send to the underloaded machine and like forage bee consecutive tasks also are sent there to Virtual Machine until the machine gets overloaded as flower patches exploitation is completed by scout bees. Honey bee behavior galvanized load balancing, improves the general turnout of process and priority based balancing focuses on reducing the makespan , time a task must help a queue of the VM. Thus, it reduces the response of your time of VMs. The experimental results show that the formula is effective when put next with existing algorithms. Our approach illustrates that there's a big improvement in average execution time and reduction in waiting time of tasks on queue. Results show that our formula stands smart while not increasing further overheads.

IX. FUTURE WORK

In future, there's scope for improvement within the algorithms. we have a tendency to arrange to improve this algorithm by considering alternative QoS factors of tasks. The performance of the given algorithms can even be augmented by variable totally different parameters. projected algorithm remains a promising and fascinating algorithm, which might still be extensively employed by researchers across various fields. Its potential advantage of being simply hybridized with totally different meta-heuristic algorithms and parts makes it robustly viable for continuing utilization for additional exploration and improvement prospects in more years to return.

REFERENCES

[1]. Kruekaew.B and Kimpan.W "Virtual Machine Scheduling Management on Cloud Computing Using Artificial Bee Colony" March 2014.

[2]. Armbrust.M, Fox.A, Griffith.R, Joseph.A.D, Katz.R, Konwinski.a, Lee.G, Patterson.D, Rabkin.A, Stoica.I, and Zaharia.M, "A view of cloud computing," in Communications of the acm, vol. 53, no. 4, April 2010, pp. 50-58.

[3]. Asaju la Aro Bolaji, Ahamed Tajudin Khader, Mohammed Azmi Al-Beter and Mohammed

[4]. Basturk.B and Karboga.D, "On the performance of artificial bee colony(ABC) algorithm," in Applied Soft Computing. 2008, pp.687-697

[5]. Calheiros.R.N, Ranjan.R, Rose.C.A.F.D and Buyya.R, "CloudSim: a toolkit for modeling and simulation of cloud computingenvironments and evaluation of resource provisioning algorithms," Software : Practice and Experience, Vol. 41, No.1, Jan. 2011, pp. 23- 50.

[6]. Hu.J, Gu.J, Sun.G, and Zhao.T, "A scheduling strategy on load balancing of virtual machine resources in cloud computing environment," in 3rd Int. Symp. on Parallel Architectures, Algorithms and Programming(PAAP), 2010, 18-20 Dec. 2010, pp. 89-96.

[7]. MainakAdhikari, SouravBanerjee, UwastpalBis, "Smart Task Assignment Model for Cloud Service Provider", Special Issue of International Journal of Computer Applications (0975 – 8887) on Advanced Computing and Communication Technologies for HPC Applications - ACCTHPCA, June 2012

[8]. Mizan.T, Masud.S.M.R.A and Latip.R, "Modified bees life algorithm for job scheduling in hybrid cloud," in Int. Journal of Engineering and Technology(IJET), 2012, vol. 2, no.6, June 2012, pp. 974-979.

[9]. Pardeep Kumar, Amandeep Verma, "Independent Task Scheduling in Cloud Computing by Improved Genetic Algorithm". Volume 2, Issue 5, May2012.

[10]. Prasanna Kumar.K, Arun Kumar.S, Dr.Jagadeeshan, "Effective Load Balancing for Dynamic Resource Allocation in Cloud Computing",in International Journal of Innovative Research in Computer and Communication EngineeringVol.2, Issue 3, March 2013.

[11]. Vahid Arabnejad, Ali Moeini and Nasrollah Moghadam, "Using Bee Colony Optimization to Solve the Task Scheduling Problem in Homogenous Systems", in Int. Journal of Computer ScienceVol. 8, Issue 5, No 3, September 2011.

[12]. Wei.Y and Tian.L "Research on cloud design resources scheduling based on genetic algorithm," in *2012 Int. Conf. on Systems and Informatics (ICSAI 2012)*, 2012, May 2012, pp. 2651-2656

[13]. Tushar Saini, Dr Ajay jangra "Scheduling Optimization in Cloud Computing", in Int. Journal of Advanced Research in Computer Science and Software Engineering(ISSN: 2277 128X) April 2013.

[14]. Bharti Mohali, Akhil Goyal "A Study of Load Balancing in Cloud Computing using Soft Computing Techniques" Int. Journal of Computer Applications (0975 8887)Volume 92 – No.9, April 2014

15 Dinesh Babu L.D, P.Venkata Krishna , "Honey Bee inspired Load Balancing of tasks in cloud computing environments", Applied soft computing 13,2292-2303,2013

16 Tejinder Sharma, Vijay Kumar Banga, "Efficient and Enhanced Algorithm in Cloud Computing",IJSCE,ISSN:2231-2301,2013

17 Joshua Samuel Raj, Hridya K.S,V. Vasudevan, "Augmenting Hierarchical Load Balancing with Intelligence in Grid Environment", International Journal of Grid and Distributed Computing Vol. 5,No. 2, pp. 9- 8, 2012

18 Guopu.Zha, Sam Kwong , "Gbest-guided Artificial bee Colony algorithm for numerical function optimization", Elsevier 3166-3173,2010

19 Erik Cuevas, Daniel Zaldívar, Marco Pérez-Cisneros, Humberto Sossa, Valentín Osuna, "Block matching algorithm for motion estimation based on ABC", Applied soft computing,2012

20 Guopu.Zha, Sam Kwong ,"Gbest-guided Artificial bee Colony algorithm for numerical function optimization", Elsevier 3166-3173,2010