International Journal of Control Theory and Applications

ISSN : 0974-5572

Volume 10 • Number 4 • 2017

Virtual Machine Availability and Load Balancing in Cloud Environment

Sakshi Arora and Sunanda

Department of Computer Science & Engineering, Shri Mata Vaishno Devi University

Abstract: Virtualization is often seen as the key to cost reduction by increasing infrastructure utilization. The main aim of the virtualization is an ability to run the multiple Virtual Machines (VMs) on a single machine by sharing all the resources that belong to the hardware. The problem of load balancing occurs when a number of users make a request to access to the same server while other servers are sitting idle. This phenomenon is called distributed load imbalance system. This issue of load imbalance can be addressed by scheduling the tasks or the services before using the system. Therefore, a good task scheduler can increase the performance of resource utilization and can reduce the makespan of assigned tasks which is called distributed load balance system. The scheduling and routing of services is based on the load of individual server and is governed by Cloud Management policies. This paper proposes an enhanced Genetic algorithm (GA)for scheduling the set of VM's so as to balance the overall load, where the makespan of the migration scheme has also to be minimized. This study performs a comparison of the average execution time of the requests with the number of requests changing; the second phase is the comparison of the average makespan with the number of VMs changing. Results indicate that the proposed GA based LBS (Load Balancer and Scheduler) makes a better utilization of the available resources in cloud.

Keywords: *Virtual Machines, Virtualization, Genetic Algorithms, Load Balancing, Task Scheduling*.

1. INTRODUCTION

Cloud computing referred to as the on demand technology because it offers dynamic and versatile resource allocation for reliable and warranted services in pay as-you-use manner to public. The resource allocation in cloud computing is nothing but integrating the cloud provider activities in order to utilize and allocate scarce resources [1]. It provides a pool of resources including virtual machines (VM) as per the requirement of the user tasks. The main objective of Cloud is to reduce costs and to provide the ease of resource management. Virtualization is often seen as the key to cost reduction by increasing infrastructure utilization. The main aim of the virtualization is an ability to run the multiple VMs on a single machine by sharing all the resources that belong to the hardware. The problem of load balancing occurs when a number of users make a request to access to the same server while other servers are sitting idle. This phenomenon is called distributed load imbalance system. This issue of load imbalance can be addressed by scheduling the tasks or the services before using the system. Therefore, a good task scheduler can increase the performance of resource utilization and can reduce the makespan of assigned tasks which is called distributed load balance

system. Task scheduling in distribution system such as Cloud is used for balancing work load. It requires some conditions for example, stability of the system; makespan of work; ability to adapt to the environment changing etc. This may lead to formation of hot and cold spots where some VMs are overloaded and others are underutilized. Load balancing techniques prove to be effective in reducing both the makespan and response time [2]. The scheduling and routing of services is based on the load of individual server and is governed by Cloud Management policies.

Distributing the resources equally promotes better resource utilization by shifting the load from heavily loaded servers to the lesser used or idle servers. The proposed Load Balancer and Scheduler (LBS)[3], [4], [5],[6] estimates parameters of a node such as the Processor speed, job arrival rate, and load on the processor for migrating jobs into lesser loaded processors.As depicted in figure 1. LBS act as a middleware between the clients, tasks trying to access processor resources and the servers to which these tasks may be mapped.

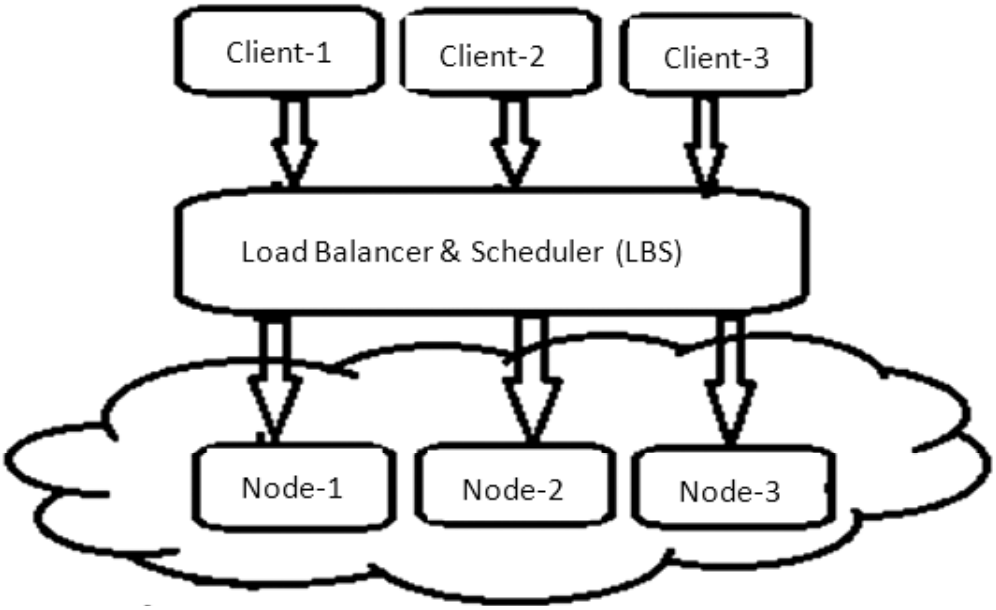


Figure 1: Load Balancer and Scheduler

The study of the proposed model considers Datacenter, Virtual Machine (VM), host and Cloud components from CloudSim for execution the algorithm. Datacenter component is used for handling service requests. VM consist of application elements which are connected with these requests. VMs are provisioned by the host components located in the Datacentre. VM life cycle starts from provisioning of a host to a VM, VM creation, VM destruction, and VM migration[7].

RELATED WORK

Several strategies have been developed for VM migration and load balancing in cloud environments. The existing works mostly have focused on the response time as an objective of load balancing. Rodrigo [8] has presented an analysis of the scalability of hosts with respect to time and memory requirements. The results indicate that the time and memory scale exponentially with the number of hosts. In [9] the server scheduling techniques in case of non preemptive tasks are compared and it is shown the priority based scheduling algorithm improves the resource utilization and reduces the response time. Warstein [10] presents VM allocation model that serves a request for VM in the minimum possible time using the least

loaded VM. In [11] LSTR strategy - a VM scheduling approach maximizing the QoS parameters is presented. Here scheduling algorithms allocate the resources among the tasks such that the cost function is maximized. MiyakoDori [12], gives a memory reusing mechanism to reduce the amount of data transferred in live VM migration. In course of task execution the Virtual machines may migrate back to the host where it was initially loaded, in such a case the memory image in that host can be reused. This strategy has shown promising results in optimizing the migration time. Kaur[13] presents an active VM load balancer algorithm to find the best VM in a shorter time duration. In case the length of the allocated VM is not sufficient then a new VM is added. The loads of the virtual machines are computed at this stage and the least loaded VM is marked for allocation to the arriving request. In [14] Ahn discusses two memory-aware cluster-level virtual machine scheduling techniques for cache sharing and nonuniform memory accesses (NUMA) Affinity. No a-priori knowledge on VM is required; instead the cloud scheduler collects the cache behavior of each VM. It is claimed that such migration may reduce the overall cache misses and the average memory access latencies by NUMA affinity. Zhong [15] presents a load balancing approach that can significantly reduce the response time with respect to the number of hosts. The algorithm has been empirically verified in cloudsim. Hu et al.[16] discusses a VM scheduling strategy using a tree structure based Genetic algorithm for load balancing. The performance of VMs is logged and the current state of the VMs in data center is used to achieve load balancing in allocating the resources. To reduce the allocation and migration schedule time Li et al. [17] proposes a Load Balancing Ant Colony Optimization (LBACO) Algorithm. ABC algorithm [18] based on Particle Swarm Optimization is used to find the most appropriate allocation within dynamic environment. A Bee Life algorithm [19] inspired by the behavior of bee to find food source has been used for scheduling in Cloud computing such as to reduce the response time. In [20] a hybrid algorithm using greedy and Bee Life approaches has been discussed.

3. LOAD BALANCING IN CLOUDS

Load balancing [21], [22], [23] in cloud environment deals with partitioning a program into tasks that can be executed concurrently and mapping each of these tasks to a processor in a manner that balances out the total load on the processors. Improved response time is the primary goal of a load balancing and improved resource/processor utilization is additionally targeted. Commercial clouds work on automatic load balancing techniques, which allow clients to increase the number of processors or to scale with application demands. Basic need of load balancing in such an environment, therefore, remains to provide the resources to the application faster and to scale with the increased resource demands of client applications [24].

Efficient provisioning of resources by means of load balancing ensures the following conditions for optimum operations in cloud environment:Resources are readily available on demand;Optimum utilization of processors and memory both under high and low load condition; Energy of the system is conserved under low load and Reduced cost of operations.

4. VM MIGRATION TECHNIQUES

Virtual machine environment is used for efficient scaling of cloud resources. AS operating system and other programs run in a VM enabled environment, the challenge is the live migration of Virtual machine in minimum possible time from one physical host to other host without disturbing others. The goal of any migration technique therefore remains reducing the total migration time and down time. The two most commonly used techniques are [25]: 1) Pre-copy: In pre-copy migration the contents of the memory are first transferred to the destination machine. After completion of this first step the processor states are transferred to the destination. 2) Post copy: In post copy technique memory contents are transferred after the processor states have been successfully transferred on to the destination machine.

5. GENETIC ALGORITHM BASED VM MIGRATION AND LOAD BALANCING ALGORITHM FOR CLOUDS

Problem Statement: Let N = {N1, N2, ..., Nn} be the set of nodes (physical machines), and n is the number of nodes on the cloud. Also let VM = {VM1, VM2, …..,VMk} be the set of virtual machines on a node.This one-to-many relationship between the physical and virtual machines is best represented as a tree structure, as shown in figure 2. The cloud controller node of the system is on the root node while all of the 'n'nodes on the second level stand for physical machines and the k nodes on the third level stand for the VMs on the physical machines. The load on a node is calculated by adding the loads of all the VM on a node.

This section presents an enhanced Genetic algorithm for scheduling the set of VM's so as to balance the overall load, where the makespan of the migration scheme has also to be minimized. In generating the optimal schedule, the Service Level Agreement (SLA) between the cloud service provider and user is consulted to reflect the requirements such as time and budget. Based on these requirements, other Qos parameters like response time are arrived at. While constructing a VM migration policy or schedule, all these parameters are to be considered. The proposed GA based Load Balancer and Scheduler (LBS) consists of the following steps:

5.1. Encoding and Initial Population Generation

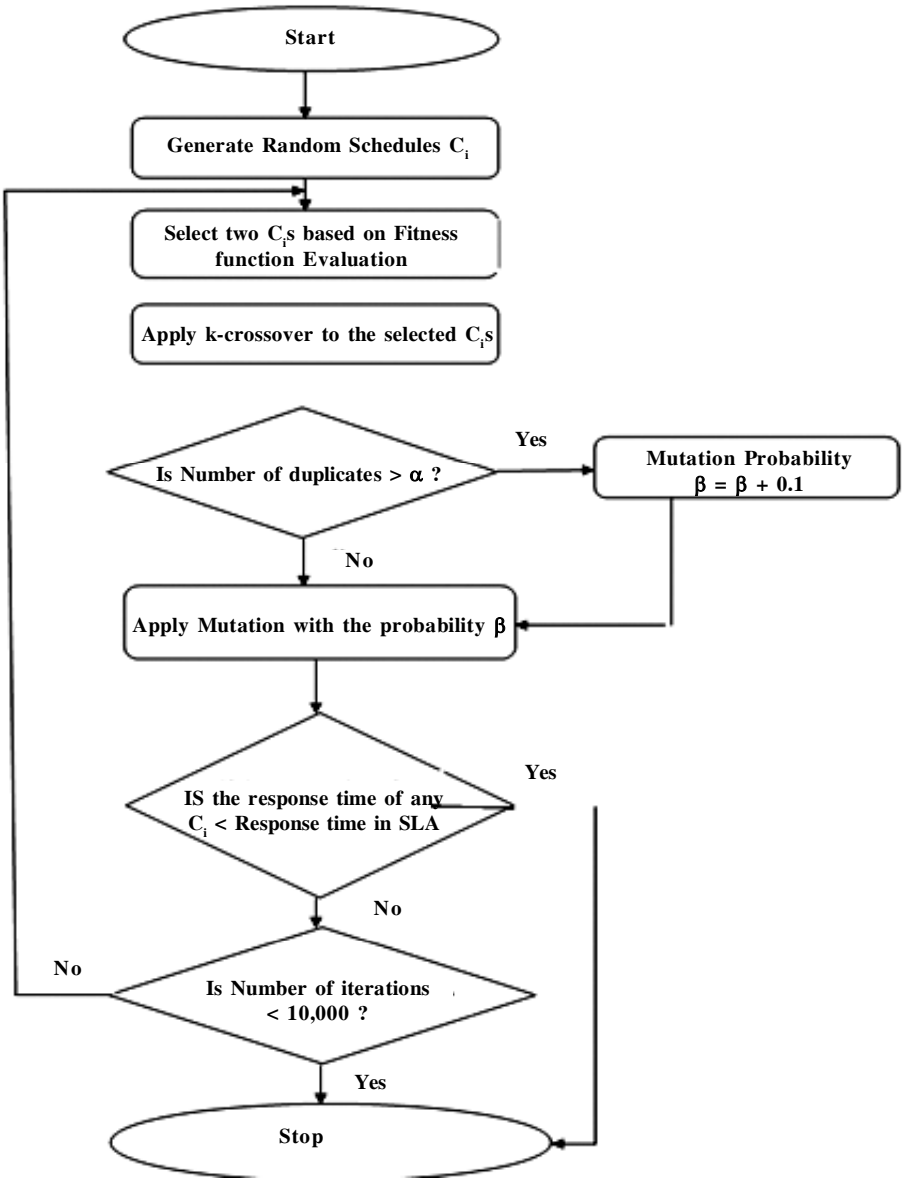
The initial set of solutions consisting of 'Max-Pop' individuals is randomly generated. A chromosome C_i in LBS indicates the assignment of tasks to specific VMs, where iranges from 1 to Max-Pop. Length of the chromosome '' is decided by the number of tasks which are inputted. This set of Inputted tasks is each assigned a priority class based on the cost. Highly paid tasks are assigned to Priority-Class 1 and are thereby allocated to the VM faster than the rest. Time constraint of the task is also considered for VM allocation.

5.2. Parent Selection

The efficiency of the Fitness function has a significant impact on the overall effectiveness of a GA based strategy in searching the solution to the problem. The Fitness value of a chromosome C_i indicates how accurate it is in allocating the VM to each task so as to minimize the overall response time. Unlike the general scheduling problems, minimization of execution time is not the only goal but also improvement in response time and throughput, where throughput can be defined as cost of processing the task. Roulette wheel selection [26] has been used for selecting the chromosomes as it is known to exert a suitable selection pressure on the search process [27]. The wheel is rotated such that the individuals with the high fitness have higher probability being selected and those with low fitness also havea chance to be chosen.

5.3. Crossover

The next step is to generate the next generation of solutions from those selected through genetic operators. k-crossover operator [28] has been used to generate the new chromosomes. This operator has been established as having high heritability quotient and therefore does not disrupt the constructed schedule entirely but creates only a slight variation of it. Mutation operator in LBS is simple bit-flip mutation. The frequency of mutation is fixed in standard in GA and is typically kept low but in LBS the frequency of the mutation operator is adjusted after each set of 20 iterations. If the number of duplicates is above (load dependent) threshold á, the frequency of mutation is increased by 0.1. This is done so as to introduce unexplored regions of the space into the search process. The dynamic adjustment of mutation operator is important in achieving the optimal solution as the heritability of the k-crossover is high, higher mutation rates particularly in later iterations can help the search process.



5.4. Termination

GA will continue iterating till the termination criterion of the required response time is met. In case the minimum response time is not reached within 10,000 iterations, the GA terminates. Fresh seeding of GA is then done using n best individuals from the current generation as in initial population to GA this time.

The flowchart of the VM scheduling and load balancing using GA based LBS is shown in Fig. 3.

6. EXPERIMENTAL SETUP

For measuring the efficiency of the proposed GA based LBS, we set up experiments on Intel(R) core(TM) i5 Processor 2.6 GHz, Windows 7 platform using CloudSim 3.0.3 simulator. The CloudSim toolkit supports modeling of components of the cloud environment such as data centers, host, virtual machines, and scheduling policies. 5 VMs were usedwith RAM of 512 MB for all Virtual Machines, and the MIPS as 250, 1000, 500, 500 and 250 respectively. Cloudlet component was used to create 15 tasks and Cloudlet length

has been set as 20000, 10000, 20000, 10000, 10000, 20000, 10000, 20000, 10000,10000, 20000 and 10000 respectively. Performance of LBS has been studied for varying number of tasks: 100, 200, 300, 400 and 500 respectively. Performance of LBS has been compared with the FCFS[29]and Round robin[30]scheduling policies.

This paper includes two experiments; the first experiment is the comparison of the average execution time of the requests with the number of requests changing, the second phase is the comparison of the average makespan with the number of VMs changing. The proposed GA based LBS makes a better utilization of the available resources in cloud is shown in the Figure 3. It reflects a graph between the number of requests as depicted on x-axis and the Resource utilization on y-axis. The graph clearly depicts better resource utilization with increasing number of requests. When the number of requests is low, the average resource utilization is not optimal but better load balancing leads to better resource utilization and hence lower execution time when the number of request are more.

Figure 4 shows the average makespan of the scheduling algorithm using the proposed GA based LBS and pitching it against Round robin and FCFS based scheduling techniques based on 50 fixed VMs and the number of increasing requests. X-axis reflects the number of tasks submitted and the y-axis depict the average makespan. The experimental results show that when the number of requests increases, the

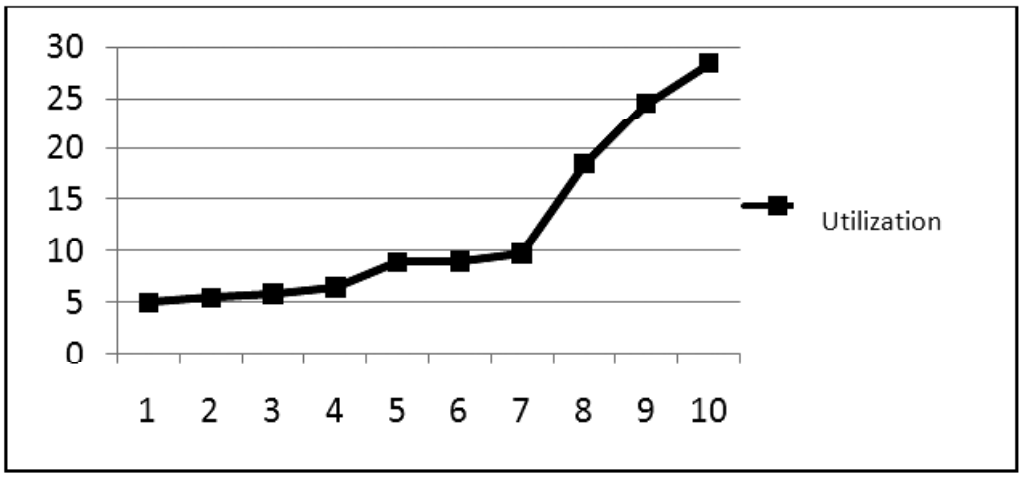


Figure 3: Average Execution Time of requests

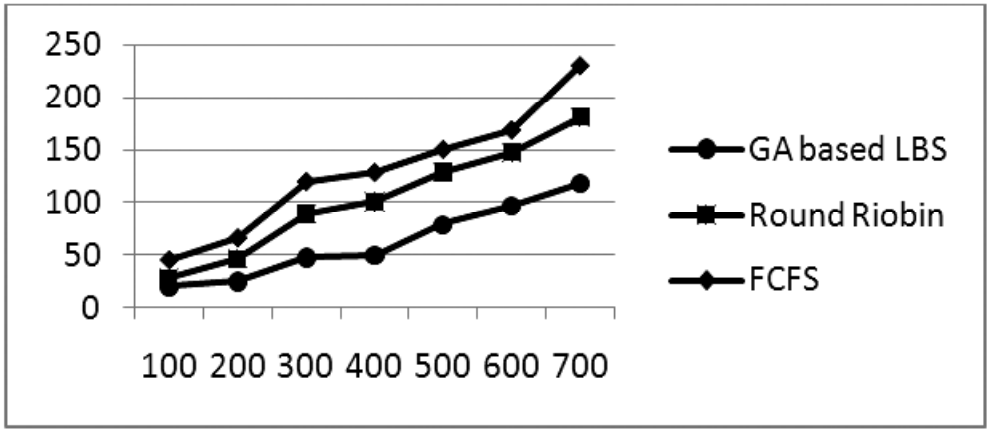


Figure 4: Comparison of average makespan among FCFS, Round robin and GA based LBS algorithm based on the fixed Number of VMs and the increased number of the requests.

average makespanscales linearly. As seen from figure 5, the proposed genetic based load balancer and scheduler, gives effective results,when compared with FCS and round robin strategies.

7. CONCLUSION

In this paper, we have proposed a load balancing and scheduling strategy based on Genetic Algorithm (LBS) for effective VM migration. The proposed strategy starts by generating random VM allocations to tasks based on the service level agreement (SLA). The individuals are then selected by applying the fitness function based on SLA requirement satisfaction and availability of the resource. K-crossover is applied to the population and the mutation rate is kept high. For evaluating the performance of LBS, we simulated the proposed model using cloudsim. Empirical results have proven that the proposed strategy outperforms existing task scheduling models, which are the round-robin task scheduling model, the FCFS scheduling model.

REFERENCES

[1] Kaur, R., &Luthra, P. (2012, December). Load balancing in cloud computing. In *Second Symposium on Cloud computing*.

[2] K. Ramana, A. Subramanyam and A. Ananda Rao, Comparative Analysis of Distributed Web Server System Load Balancing Algorithms Using Qualitative Parameters, VSRD-IJCSIT, Vol. 1 (8), 2011, 592-600.

[3] M. Beltran, A. Guzman and J.L. Bosque(2011), "Dealing withheteroginity in clusters" in proceeding of the Fifth International Symposium on Parallel and Distributed Computing, ISPDC.

[4] P. Warstein, H. Situ and Z. Huang(2010), "Load balancing in a cluster computer" In proceeding of the seventh International Conference on Parallel and Distributed Computing, Applications and Technologies, IEEE.

[5] Zenon Chaczko, Venkatesh Mahadevan, Shahrzad Aslanzadeh, Christopher Mcdermid (2011) "Availabity and Load Balancing in Cloud Computing" International Conference on Computer and Software Modeling IPCSIT vol. 14 IACSIT Press,Singapore 2011.

[6] Zhong Xu, Rong Huang,(2009) "Performance Study of Load Balanacing Algorithms in Distributed Web Server Systems", CS213 Parallel and Distributed Processing Project Report.

[7] Bhathiya, Wickremasinghe. (2010) "Cloud Analyst: A Cloud Sim-based Visual Modeller for Analysing Cloud Computing Environments and Applications".

[8] Calheiros Rodrigo N., Rajiv Ranjan, César A. F. De Rose, RajkumarBuyya (2009): Cloud Sim: A Novel Framework for Modeling and Simulation of Cloud Computing Infrastructures and Services CoRR abs/0903.2525: (2009).

[9] FatosXhafa, Ajith Abraham, "Computational models and heuristic methods for Grid scheduling problems", "Future Generation Computer Systems 26", 2010, pp. 608-621.

[10] P. Warstein, H. Situ and Z. Huang (2010), "Load balancing in a cluster computer" In proceeding of the seventh International Conference on Parallel and Distributed Computing, Applications and Technologies, IEEE.

[11] David W Chadwick*, Matteo Casenove ,Kristy Siu "My private cloud – granting federated access tocloud resources" Chadwick et al. Journal of Cloud Computing: Advances, Systems and Applications 2013,Springer.

[12] Soramichi Akiyama, Takahiro Hirofuchi, Ryousei Takano, Shinichi Honiden (2012), "MiyakoDori: A Memory Reusing Mechanism for Dynamic VM Consolidation", Fifth International Conference on Cloud Computing, IEEE 2012.

[13] Jaspreet kaur (2012),"Comparison of load balancing algorithms in a Cloud" International Journal of Engineering Research and Applications(IJERA) ISSN: 2248-9622 www.ijera.comVol. 2, Issue 3,pp.1169-1173.

[14] JeongseobAhn, Changdae Kim, Jaeung Han, "Dynamic Virtual Machine Scheduling in Clouds for Architectural Shared Resources".

[15] Zhong Xu, Rong Huang,(2009)"Performance Study of Load Balanacing Algorithms in Distributed Web Server Systems", CS213 Parallel and Distributed Processing Project Report.

[16] Hu, Gu, G. Sun, and T. Zhao, "A Scheduling Strategy on Load Balancing of Virtual Machine Resources in Cloud Computing Environment", Third International symposium on parallel architecture, algorithms and programming(PAAP),pp.89-96,2010.

[17] Li K, Xu G, Zhao G, Dong Y, Wang D. Cloud task scheduling based on load balancing ant colony optimization. Sixth AnnuChinagridConf 2011;2011:3–9. http://dx.doi.org/10.1109/ ChinaGrid.2011.17.

[18] M. A. Tawfeek, A. El-Sisi, A. E. keshk and F. A. Torkey, " Artificial Bee Colony Algorithm for Cloud Task Scheduling", International Journal of Computer and Information (IJCI), vol. 4, no. 1, pp. 1-9, 2015.

[19] S. Bitam, "Bees life algorithm for job scheduling in cloud computing," in Conf. on Computing and Information Technology (ICCIT 2012), 2012, pp. 186-191.

[20] T. Mizan, S. M. R. A. Masud, and R. Latip, "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.

[21] GaochaoXu,Junjie Pang, and Xiaodong Fu, Tsinghua "Load balancing model based on Cloud partitioning for the public cloud", science and technology 2013,vol.18, pp34-39.

[22] Chun-Cheng Lin,Hui-Hsin Chin, and Der-Jiunn Deng, IEEE members, "Dynamic Multiservice Load balancing in Coud –Based Mutimedia System", IEEE Systems Journal,2013.

[23] Hung Chang, Hsueh-Yi Chung, Haiying Shen and Yu-Chang Chao , " Load Rebalancing for Distributed File System in clouds", IEEE Transactions on Parallel And Distributed Systems.2013, vol.24, pp951-961.

[24] Zenon Chaczko, VenkateshMahadevan, ShahrzadAslanzadeh, Christopher Mcdermid (2011)"Availabity and Load Balancing in Cloud Computing" International Conference on Computer and Software Modeling IPCSIT vol.14 IACSIT Press,Singapore 2011.

[25] Getzi Jeba Leelipushpam. P, "Live Virtual Machine Migration Techniques – A Survey" in International Journal of Engineering Research & Technology (IJERT), Vol. 1 , September – 2012.

[26] Pop F, Dobre C, Cristea V. Genetic algorithm for DAG scheduling in grid environments. In: 5th IEEE intconfintellcomputcommun process; 2009. p. 299–305.

[27] Back, Thomas. "Selective pressure in evolutionary algorithms: A characterization of selection mechanisms." *Evolutionary Computation, 1994. IEEE World Congress on Computational Intelligence., Proceedings of the First IEEE Conference on*. IEEE, 1994.

[28] Arora, Sakshi. "Heuristic and metaheuristic solutions for the bounded diameter Minimum Spanning tree."Ph.D. thesis, 2014. shodhganga.inflibnet.ac.in/handle/10603/36688.

[29] Er. Shimpy, Jagadeep Sidhu "Different Scheduling Algorithm in different cloud environment" IJARCCE Vol 3, issue 9, September 2014.

[30] Xiaonian Wu, Mengqing Deng , Runlian Zhang , Bing Zeng, Shengyuan Zhou "A task Scheduling algorithm based on QOSdriven" International conference on Information Technology and quantitative Management(ITQM) pp. 1162-1169 , ELSEVIER(2013).