



Improving Load Balancing using Dynamic Algorithms in Cloud Environment

Ms. Hemali S. Jinjuwadia
PG Student of Computer Science Department
B.H. Gardi College of Engineering & Technology,
Rajkot, Gujarat, India.

Prof. Shyam R. Kotecha
Asst. Professor: CSE Department
B.H. Gardi College of Engineering & Technology,
Rajkot, Gujarat, India.

Abstract - Cloud computing is a promising paradigm which provides resources to customers on their request with minimum cost. Cost effective scheduling and load balancing are major challenges in adopting cloud computation. Features such as elasticity, scalability, universal access, low entry cost, and flexible billing motivate consumers to migrate their core businesses to the cloud.

Load Balancing is the one of the most important parts of the current virtual environment. In the case of cloud computing environments there were various challenges are there in the load balancing techniques like security, any fault tolerance etc. Efficient load balancing methods avoids under loaded and heavy loaded conditions in data centers. The goal of this paper is to help in developing a new algorithm after studying almost all available algorithms. When some VMs are overloaded with several number of tasks, these tasks are migrated to the under loaded VMs of the same datacenter in order to maintain Quality of Service (QoS). The goal of this paper is to help in developing a new algorithm after studying almost all available algorithms.

Keywords: Static and dynamic load balancing, Cloud computing, Task scheduling, algorithm related to dynamic, comparative study using parameters.

I. INTRODUCTION

Cloud Computing is a concept that has many computers interconnected through a real time network like a connecting a computer to any other computer anywhere in the world via dedicated routers and servers.[1] To handle a very large amount of data several techniques to optimize load and streamline operations are needed to achieve desired performance level for the users.[2]

Load balancing helps to distribute all loads between all the nodes. It also ensures that every computing resource is distributed efficiently and fairly. It helps in preventing bottlenecks of the system which may occur due to load imbalance. It provides high satisfaction to the users. Load balancing is a relatively new technique that provides high resource utilization and better response time.[3]

The dynamic nature of cloud computing environment needs a dynamic algorithms for efficient and efficient scheduling and load balancing among nodes. Static load balancing algorithms will works only when small variation in the workloads. [4]. In dynamic environment the cloud providers installs heterogeneous resources. The resources are flexible in dynamic environment. In this scenario cloud cannot rely on the prior knowledge whereas it takes into account run-time statistics.[5] The main problem is to allocate load in distributed manner to different node of the whole system so that the computation is completed in the optimum possible time.[2]

II. LITERATURE SURVEY ON LOAD BALANCING

In a cloud environment, by using various load balancing techniques improvement in resource utilization, job response time can be measured. An ideal load balancing algorithm avoids overloading or under loading of any specific node [2]. In [6], a **load balancing approach** was discussed, which manages load at server by considering the current status of all available VMs for assigning the incoming requests. This VM-assign load balancing technique mainly considers efficient utilization of the resources and VMs. By simulation, they proved that their algorithm distributes the

load optimally and hence avoids under/over utilization of VMs. The comparison of this algorithm with active-VM load balance algorithm shows that their algorithm solves the problem of inefficient utilization of the VMs.

Response time based load balancing is presented in [7]. In order to decide the allocation of new incoming requests, proposed model considers current responses and its variations. The algorithm eliminates need of unnecessary communication of the Load Balancer. This model only considers response time which is an easily available with the Load Balancer as each request and response passes through the Load Balancer, hence eliminates the need of collecting additional data from any other source thereby wasting the communication bandwidth. In [8] a load balancing technique for cloud datacenter, Central Load Balancer (CLB) was proposed, which tried to avoid the situation of over loading and under loading of virtual machines. Based on priority and states, the Central Load Balancer manages load distribution among various VMs. CLB efficiently shares the load of user requests among various virtual machines.

A. Matrices in the load balancing

Table 1. Matrices in the Load balancing [10]

Metric	Illustration
Through Put	Throughput It is used to calculate the no. of tasks whose execution has been completed. It should be high to improve the performance of the system
Overhead	It determines the amount of overhead involved while implementing a load balancing algorithm. It is composed of overhead due to movement of tasks, Inter-processor and inter-process communication. This should be minimized so that a load balancing Technique can work efficiently.
Fault Tolerance	It is the time to migrate the jobs or resources from one node to other. It should be minimized in order to enhance the performance of the system.
Response Time	It is the amount of time taken to respond by a particular load balancing algorithm in a distributed system. This parameter should be minimized.
Resource Utilization	It is used to check the utilization of resources. It should be optimized for an efficient load balancing.
Scalability	It is the ability of an algorithm to perform load balancing for a system with any finite number of nodes. This metric should be improved.
Performance	It is used to check the efficiency of the system. This has to be improved at a reasonable cost, e.g., reduce task response time while keeping acceptable delays.

B. Challenges in the load balancing

Some scientific challenges still remain unsolved by the scientific community. Some main challenges in load balancing are following:[11]

1) Automated Service Provisioning:-

A key feature of cloud computing is elasticity i.e., resources can be allocated or released automatically. Then how can we use or release the resources of the cloud, by keeping the same performance as traditional systems and using optimal resources?

2) Virtual Machine Migration:-

Virtualization makes it possible to see an entire machine as a file or set of files. Virtualization also provides a facility to move a virtual machine among heavily loaded physical machines so that balance can be achieved among them. The main objective of virtual machine migration is to distribute the load in a datacenter or set of datacenters. Then how can we dynamically distribute the load when moving the virtual machine to avoid bottlenecks in Cloud computing systems?

3) Energy Management:-

Energy saving is a main challenge for load balancing algorithms. Energy saving is extremely needed in cloud environment to achieve green computing. There is always a need of energy efficient algorithm which minimizes resource consumption but keeps acceptable performance.

4) Stored Data Management:-

In the last few years exponential growth has been seen in stored data across networks. This data may belong to any company or any individual. The management of data storage for companies or individuals, become a major challenge for cloud computing. Then how can we distribute the data to the cloud for optimum storage of data while maintaining fast access?

5) Emergence of Small Data Centres:-

Small datacenters have some benefits over large data centers like they are more beneficial, cheaper and less energy consumer. Small providers can deliver cloud computing services leading to geo-diversity computing. Load balancing will become a problem on a global scale to ensure an adequate response time with an optimal distribution of resources.

C. Types of Load Balancing

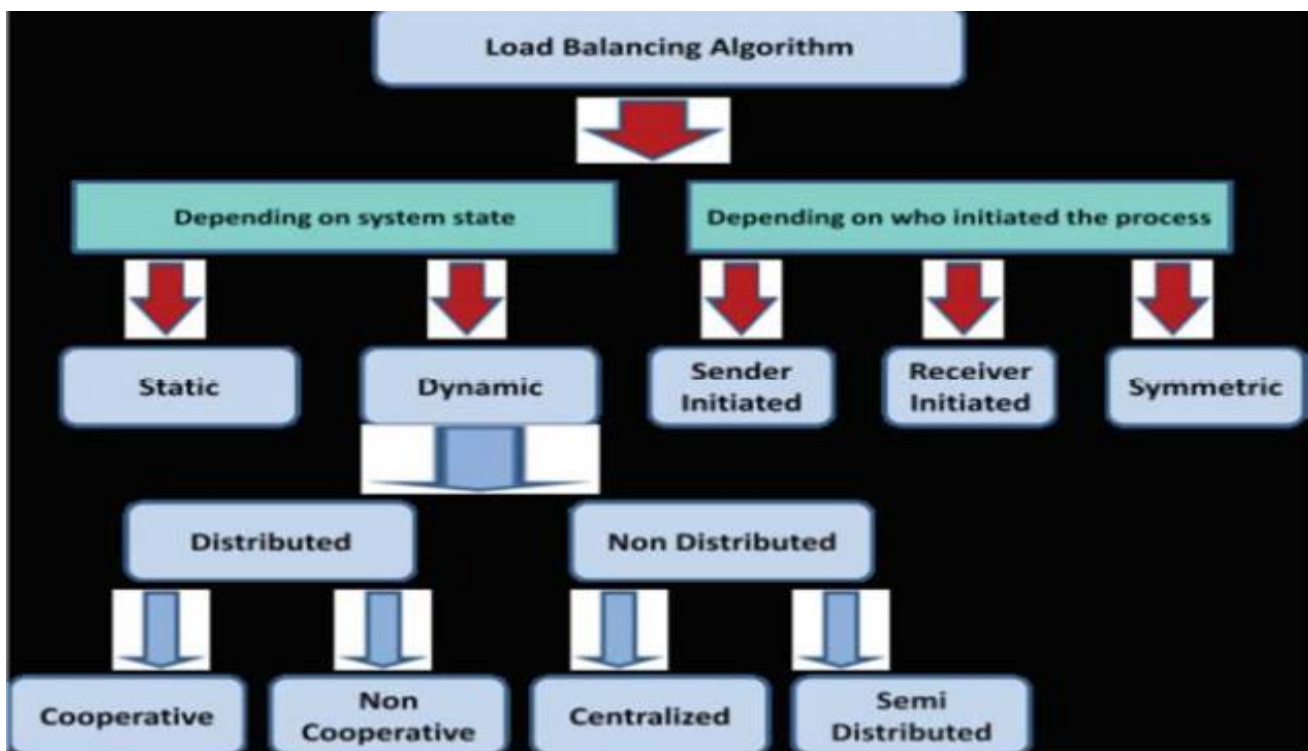


Figure 1. Types of Load Balancing Algorithm [9]

Load balancing algorithms are broadly divided into two categories:-

1) **Depending on system state:-**

In this type of load balancing technique either the load is work as under loaded or over loaded that is decided on the basis of static and dynamic.

1.1) Static:-

In static environment the cloud provider installs homogeneous resources. Also the resources in the cloud are not flexible when environment is made static. In this scenario, the cloud requires prior knowledge of nodes capacity, processing power, memory, performance and statistics of user requirements. These user requirements are not subjected to any change at run-time. Algorithms proposed to achieve load balancing in static environment cannot adapt to the run time

changes in load. Although static environment is easier to simulate but is not well suited for heterogeneous cloud environment.[5]

1.2) Dynamic:-

These algorithms take decisions concerning load balancing based upon the current state of the system and don't need any prior knowledge about the system. This approach is an improvement over the static approach. The algorithms in this category are considered complex, but have better fault tolerance and overall performance. [12]

1.2.1) Distributed:-

Dynamic load balancing algorithm is executed by all nodes present in the system and the task of scheduling is shared among them.[1] In distributed load balancing technique, no single node is responsible for making resource provisioning or task scheduling decision.[5]

1.2.1.1) Cooperative:-

The common goal of this type of algorithm is to optimize the response time [2]. In cooperative all the elements of load balancing are working in cooperative manner.

1.2.1.2) Non- Cooperative:-

All running tasks are independent of each other and thus improving the total response time for the local task [2]. None of the task is sending or receiving any update from other tasks.

1.2.2) Non-Distributed:-

In the non-distributed or undistributed, the nodes work Personal in order to instate a common goal. Non-distributed dynamic load balancing algorithms are ahead Classified into two: centralized and semi -centralized. [1]

1.2.2.1) Centralized:-

In centralized load balancing technique all the allocation and scheduling decision are made by a single node. This node is responsible for storing knowledge base of entire cloud network and can apply static or dynamic approach for load balancing. This technique reduces the time required to analyze different cloud resources but creates a great overhead on the centralized node. [10]

1.2.2.2) Semi-Distributed:-

In semi-distributed dynamic load balancing, the nodes of the system are divisions into clusters, where the load balancing in each cluster is of centralized form. A central node is elected in every cluster by appropriate election technique which takes care of load balancing within that cluster.

2) Depending on who initiate the process: --

Based on the process initiation the load balancing types are divided into three parts as follows. [11]

2.1) Sender Initiated:-

In this algorithm firstly client sends request and then a receiver is assign to him to receive his workload i.e. the sender initiates the process.

2.2) Receiver Initiated:-

In this algorithm firstly receiver sends a acknowledged request to a sender who is prepared to share the workload i.e. the receiver initiates the process.

2.3) Symmetric:-

It is a combination of both sender and receiver initiated type of load balancing algorithm.

III. DYNAMIC LOAD BALANCING ALGORITHM

Dynamic load balancing algorithms make any decision for load balancing based on dynamically changing state of the system. It allows for processes to move from an over utilized machine to an underutilized machine dynamically for any faster execution. This means that dynamic load balancing is pre-emptive which helps in improving the overall performance of the system by migrating the load dynamically. [14]

When the work load is distributed among the processor at runtime then it is known as dynamic load balancing. In that mechanism, master assigns new processes to the slaves based on the new information collected. As in dynamic load balancing algorithm each nodes of the system have to communicate with every other node of system, it will generate more messages. In centralized dynamic algorithm number of messages for interactions are decreases drastically as compared to the semi distributed case and so it reach to a decision very fast. [2]

Following are some load balancing algorithm that provides response time as well as performance improvement in cloud computing:-

A. The Ant Colony Optimization:-

The main idea concluded from ACO is to simulate the foraging behaviour of ants that try to search for the abundant food sources. They exploit a special kind of chemical pheromone to communicate with each other. Initially ants start looking for their foods randomly and when they detect a path to food source, they drop a specified amount of pheromone on this path. Other ants can go ahead to the food source by sensing pheromone on the ground. This process continues until most of ants are enticed to select shortest path that contains huge amount of pheromones collected on this path. The process of ACO algorithm is the use of the positive feedback mechanism and inner parallelism. The cons are overhead and the stagnation phenomenon that make the algorithm converging to local optimal solution [15]. In the Cloud based to provide dynamical load balance, ACO is used to find the near-optimal resource allocation for dynamic tasks in the cloud to minimize the make span of tasks.

B. Particle Swarm Optimization:-

PSO has been invented in 1995. Particles in the swarm fly over an environment following the best members of the swarm and directing their movement toward good areas from their environment. Initially, this algorithm assigns random positions to all particles in search space. It advances the position of each particle successively based on its velocity using the global best known position and the best position known to a particle. Over time, the particles get together around optima or several optimal. The algorithm keeps track of global best known position (gBest) and stopping value indicating when the algorithm should stop. Each particle consists of data that represent a solution, personal best (pBest) value and velocity value denoting how much the data can be modified. The velocity value is calculated according to how particle's data is out of the way from the target. The further particle needs the high amount of velocity value. As in the bird's example, the individuals outmost from the food would carry out an effort to follow with the others by flying faster toward the best bird. PSO is instantiated with a specific number of particles. Each particle concerns one potential solution to the faced scheduling problem. The solution is represented as an array of VM's IDs representing the order of VMs. [15]

C. Artificial Honey Bee Colony Algorithm:-

It is the method to find the appropriate value. This method is inspired by the foraging behaviour of honey bees. In ABC model, there are three kinds of honey bee to search food sources, which include scout bees search for food source randomly, employed bees search around the food source and share food information to the onlooker bees, and onlooker bees calculate the fitness and select the best food source. Cloud computing provides a dynamic resource pool of VM according to different requirements from the users or the system. The routing of services which request to the diverse servers, depend on the Cloud management policies based on load of individual server. [16]

D. Honey Bee Foraging Algorithm:-

This is nature inspired and self-organizing algorithm used to solve load balancing in dynamic cloud environment. Honeybees have ability to find their food and inform other bees for destination food in the bee's colony. Each forager bees find their food initially then go back to inform other bees. Forager bees used different dancing movements for location and direction of food stuffs. When bees found more food more energetic dance represents to scout bees then it follow forager bees and get food. This phenomena applied on overloaded and under loaded virtual server. When clients request to server if found overloaded it redirect request to other under loaded virtual server. [2] Honey bees have been classified into two

types. They are: finders and reapers. The finder honeybee helps in finding the honey source. Once honey source is found, they do the waggle dance to indicate the quality and quantity of available honey. After that, the reapers gather the honey from the sources. Then, again they go for the waggle dance to specify the honey that is left. In load balancing, the servers are combined together as virtual servers, where each and every virtual server has a process queue. Once the request is received from the queue, it calculates the profit quality as the bee does in waggle dance. The server stays only when the profit is high, or else proceeds to forage by indicating that whether the state is loaded, overloaded, under loaded or balanced. Based on this, the current virtual machines are combined. It needs to maintain a separate queue for each and every node. Depending on the priority, the task is taken into concern, by removing the task that is waiting in the overloaded machine. The tasks removed are loaded into lightly loaded machine. Those tasks are known as scout bee for the next step. The behaviour of honey bee in load balancing technique has stimulated to reduce the response time of virtual machine, which also reduces the waiting time. The main disadvantage of this algorithm is, it does not show any improvement in throughput. [17]

E. Central queuing:-

This algorithm works on the principal of dynamic distribution. Each new activity arriving at the queue manager is inserted into the queue. When request for an activity is received by the queue manager it removes the first activity from the queue and sends it to the requester. If no ready activity is present in the queue the request is buffered, until a new activity is available. But in case new activity comes to the queue while there are unanswered requests in the queue the first such request is removed from the queue and new activity is assigned to it. When a processor load falls under the threshold then the local load manager sends a request for the new activity to the central load manager. The central manager then answers the request if ready activity is found otherwise queues the request until new activity arrives. [18]

F. Least Connection mechanism based algorithm:-

Load balancing algorithm can also be based on least connection mechanism which is a part of dynamic scheduling algorithm. It needs to count the number of connections for each server dynamically to estimate the load. The load balancer records the connection number of each server. The number of connection increases when a new connection is dispatched to it, and decreases the number when connection finishes or timeout happens.[18]

G. Biased Random Sampling Load Balancing Algorithm:-

Biased Random Sampling is a dynamic load balancing algorithm. Here, random sampling method is being used to achieve the load balancing across all the nodes. In this algorithm, all the servers are treated as nodes. This method is represented in the form of virtual graph, constructed with the connectivity which represents the load on each node. Each node is taken as vertex in a directed graph. When a request is received from the client to the load balancer, the load balancer assigns the job to the node that has a minimum of one in- degree. Once a job is assigned to the node, the server starts executing the job, indicating the reduction in availability of free resources. After the completion of the job, the node gets incremented by one in-degree, indicating the increase in available resources. The addition and deletion of such processes are completed by the process of random sampling technique. Threshold value is used as a parameter that considers each and every process by representing the maximum walk length. The traversal is from one node to another node until finding a designation is known as a walk. After receiving the request from the load balancer, it compares the current node to the randomly selected node with the threshold value. If the threshold value is equal or greater than the current walk length, the node executes its job, or else it moves to another neighbour node that is randomly selected. The performance decreases as the number of servers increases. [17]

H. Active Clustering Load Balancing Algorithm:-

Active Clustering is an improved method of random sampling. The concept of clustering is used in this algorithm. The main principle of this algorithm is grouping similar nodes together, and working based on those grouped nodes. Grouping of nodes helps the resources to increase the throughput efficiently. In this algorithm, a method called match-maker is introduced. While an execution starts, the first node selects the neighbour node. The neighbour node is taken as match make node, which connects the neighbour node that is same as initial node. At last the match maker node gets disconnected. And this process is done iteratively to balance the load equally. The system performance is improved highly, by increasing the throughput. There is an efficient utilization of resources when there is an increase in throughput [2]. In this algorithm same type nodes of the system are grouped together and they work together in groups. It works like as self-aggregation load balancing technique where a network is rewired to balance the load of the system. Systems optimize using similar job assignments by connecting similar services. System Performance improved with improved resources. The throughput is improved by using all these resources effectively. [9]

I. Join-Idle-Queue:-

This algorithm provides large-scale load balancing with distributed dispatchers by, first load balancing idle processors across dispatchers for the availability of idle processors at each dispatcher and then, assigning jobs to processors to reduce average queue length at each processor. A Join-Idle-Queue load balancing algorithm for dynamically scalable web services. It effectively reduces the system load, incurs no communication overhead at job arrivals and does not increase actual response time. It can perform close to optimal when used for web services. However, it cannot be used for today's dynamic-content web services due to the scalability and reliability. [19] It first assigns idle processors to dispatchers for the availability of the idle processors at each dispatcher and then assigns jobs to processors to reduce average queue length of jobs at each processor. [14]

J. A Fast adaptive load balancing method:-

A binary tree structure that is used to partition the simulation region into sub-domains. The characteristics of this fast adaptive balancing method. The characteristics of this fast adaptive balancing method are to be adjusted the workload between the processors from local areas to global areas. According to the difference of workload, the arrangements of the cells are obtained. But the main workload concentrates on certain cells so that the procedure of adjusting the vertices of the grid can be very long because of the local workload can be considered. This problem can be avoided by the fast load balancing adaptive method. Here the region should be partitioned by using the binary tree mode, so that it contains leaf nodes, child nodes, parent nodes etc. There were partition line between the binary tree and the indexes of the cells on the left are smaller that of right and the indexes on the top are smaller than the bottom. Calculate the workload based on the balancing algorithm. This algorithm has a faster balancing speed, less elapsed time and less communication time cost of the simulation procedure. Advantages are relative smaller communication overhead, faster balancing speed, and high efficiency and the disadvantage is it cannot maintain the topology that is neighbouring cells cannot be maintained. [10]

K. Heat Diffusion Based Dynamic Load Balancing:-

In this algorithm, proposed an efficient cell selection scheme and two diffusion based algorithm called global and local diffusion. According to heat diffusion algorithm, the virtual environment is divided into a large no of square cells and each square cell having object and every node in the cell send load to its neighbouring nodes in every iteration and the transfer was the difference between the current node to that of neighbouring node. it is related to heat diffusion process. The advantages of this algorithm are communication overhead is less, high speed and require little amount of calculation. [20] Considered the distributed virtual environments there were various numbers of users and the load accessing by the concurrent users can cause problem. This can be avoided by this algorithm. According to the heat diffusion algorithm; the virtual environment is divided in two large numbers of square cells and each square cell having objects. [10]

L. Local Queue Algorithm:-

Main feature of Local Queue algorithm is dynamic process migration support. Basically it supports static allocation of all new processes with process migration which is initiated by a host when it is under loaded. The load capacity is defined by a user defined parameters. The parameter defines how many minimum ready processes the load manager can attend. When a new process is created on the main host it will be allocated on under loaded hosts. As the first parallel construct of activities is sufficient for allocation of load to all remote hosts, all other processes created on the main host and other hosts then after are allocated locally. When a host is under loaded it randomly sends request with the number of local ready processes to remote load managers. When a load manager receives this type of request, it will compare the local ready processes with the received number and if the local processes are grater then some running processes will be transferred to the requester. The requesting node will send a confirmation to the sender after receiving the processes. [2]

IV. QUALITATIVE PARAMETERS

There are so many parameters like nature, response time, resource utilization, fault tolerance, throughput, reliability, scalability etc are to be considered in the dynamic load balancing algorithm which are described as follows:-

Table 2. Comparative Analysis of Qualitative parameters in Dynamic Algorithm

Parameters	Central Queue	Local Queue	Ant Colony	Particle Swarm	Artificial Bee Colony	Honey Bee Foraging	Active Clustering	Biased Random Sampling
Nature	Dynamic	Dynamic	Dynamic	Dynamic	Dynamic	Dynamic	Dynamic	Dynamic
Response time	More	More	More	More	More	More	Less	Less
Resource Utilization	Less	More	More	More	More	More	More	Less
Fault Tolerance	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Waiting Time	Less	Less	Less	Less	Less	Less	More	More
Reliability	More	More	More	More	More	More	More	More
Throughput	High	High	High	High	High	High	High	Low
Turn Around Time	More	More	More	More	More	More	Less	Less
Adaptability	More	More	More	More	More	More	More	Less
Stability	Small	Small	Small	Small	Small	Small	Large	Large
Process Migration	No	Yes	Yes	Yes	Yes	Yes	No	No
Cooperative	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Predictability	Less	Less	Less	Less	Less	Less	Less	More
Process Thrashing	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

V. CONCLUSION

In this Paper, there are general terms and ideas related to load balancing were discussed, so we can able to understand what actually the load balancing in cloud, what are the challenges, issues, matrices etc of load balancing. Here many algorithms related with dynamic load balancing were explained only because dynamic load balancing in cloud is more dynamic than static as well as dynamic load balancing has more qualitative parameters than the static one. That is the reason to move towards dynamic approach.

In future work, we need to implement all the dynamic algorithm and from them find and choose the better one out of all that can provide good load balancing. According to analysis of all the above mentioned algorithm we consider honey bee as a future work algorithm because in this mainly the response time and spanning are reduced. So using Honey bee we can try to reduce the response time in load balancing. Our goal is to reduce the response time using honey bee as well as try to use hybridization between ACO and PSO and then by doing comparison we can able to conclude for better one.

REFERENCES

- [1] Vinza V. Suthan, Chithranjan K.: Survey On Load Balancing in cloud Computing, In: International Journal of Computer and Advanced Engineering Research (IJCAER), ISSN 2395 – 4523 Volume 02– Issue 02, APRIL 2015
- [2] Ms. Kunjal Garala, Ms. Namrata Goswami: A performance Analysis of Load balancing Algorithms in cloud Environment, In: International Conference on Computer Communication and Informatics (ICCCI -2015), Jan. 08 – 10, 2015, Coimbatore, INDIA

- [3] Rajwinder Kaur, Pawan Luthra: Load Balancing in Cloud Computing In: Proc. of Int. Conf. on Recent Trends in Information, Telecommunication and Computing, ITC, ACEEE
- [4] K.R. Ramesh Babu, Philip Samuel: Enhanced Bee Colony Algorithm for Efficient Load balancing and Scheduling in cloud, In: Springer International Publishing Switzerland 2016 V. Snasel et al. (eds.), Innovations in Bio-Inspired Computing and Applications, Advances in Intelligent Systems and Computing 424
- [5] Mayanka Katyal, Atul Mishra.: A Comparative Study of load balancing algorithm in cloud environment. In: International Journal of Distributed and Cloud Computing Volume 1 Issue 2 December 2013
- [6] Wang, L., Zhou, G., Xu, Y., Liu, M.: An enhanced Pareto-based artificial bee colony algorithm for the multi-objective flexible job-shop scheduling. Int. J. Adv. Manuf. Technol. **60** (Issue 9–12), 1111–1123. Springer (2012)
- [7] Domanal, S.G.R., Ram Mohana, G.: Load balancing in cloud computing using modified throttled algorithm. In: IEEE International Conference on Cloud Computing in Emerging Markets (CCEM), pp. 1–5 (2013)
- [8] Shridhar, G.D., Reddy, G.R.M.: Optimal load balancing in cloud computing by efficient utilization of virtual machines. In: IEEE Sixth International Conference on Communication Systems and Networks (COMSNETS), pp. 1–4 (2014)
- [9] Sharma, A., Peddoju, S.K.: Response time based load balancing in cloud computing. In: International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT), pp. 1287–1293 (2014)
- [10] Sukhvir Kaur¹, Supriya Kinger.: Review on load balancing technique in cloud environment. In: International journal of Science and Research (IJSR), Volume 3 Issue 6, June 2014
- [11] Nitin Kumar Mishra, Nischol Mishra.: Load Balancing Techniques: Need, Objectives and Major Challenges in Cloud Computing- A Systematic Review, In: International Journal of Computer Applications (0975 – 8887) Volume 131 – No.18, December 2015
- [12] N.S.Raghva and Deepti Singh: Comparative study on load balancing techniques in cloud computing, In: Open journal of mobile computing and cloud computing Volume 1, Number 1, August 2014
- [13] Ram Prasad Padhy, P Goutam Prasad Rao : Load balancing in cloud computing System, In: Department of Computer Science and Engineering National Institute of Technology, Rourkela, May, 2011.
- [14] Mr. M. Ajit, Ms. G. Vidya. VM level load balancing in cloud environment, In: IEEE Journal, 4th ICCCNT 2013 July 4-6, 2013
- [15] Gamal F. Elhady and Medhat A. Tawfeek.: A Comparative Study into Swarm Intelligence Algorithms for Dynamic Tasks Scheduling in Cloud Computing. In: IEEE Seventh International Conference on Intelligent Computing and Information Systems (ICICIS'15)
- [16] B. Kruekaew and W. Kimpan: Virtual Machine Scheduling Management on Cloud Computing Using Artificial Bee Colony. In: Proceedings of the International Multi Conference of Engineers and Computer Scientists 2014 Vol. I, IMECS 2014, March 12 - 14, 2014
- [17] D. Saranya, L. Sankara Maheswari.: International Journal of Advanced Research in Computer Science and Software Engineering, Volume 5, Issue 7, July 2015
- [18] Soumya Ray and Ajanta De Sarkar.: Execution Analysis Of Load Balancing Algorithms In Cloud Computing Environment In: International Journal on Cloud Computing: Services and Architecture (IJCCSA), Vol.2, No.5, October 2012
- [19] Nayandeep Sran, Navdeep Kaur.: Comparative Analysis of Existing Load Balancing Techniques in Cloud Computing In: International Journal of Engineering Science Invention.
- [20] Amandeep, Vandana Yadav, and Faz Mohammad.: Different Strategies for Load Balancing in Cloud Computing Environment: a critical Study. : In: International Journal of Scientific Research Engineering & Technology (IJSRET), Volume 3 Issue 1, April 2014