

# Dynamic Load Balancing Algorithms For Cloud Computing

Miss. Nikita Sunil Barve  
Computer Engineering Department  
Pillai's Institute of Information Technology  
New Panvel  
*e-mail: niki.barve@gmail.com*

Prof. Manjusha Deshmukh  
Computer Engineering Department  
Pillai's Institute of Information Technology  
New Panvel  
*e-mail: mdeshmukh@mes.ac.in*

Miss . Prgya Tripathi  
Computer Engineering Department  
Pillai's Institute of Information Technology  
New Panvel  
Email: 123pragya.tripathi@gmail.com

**Abstract**— In cloud computing, the load balancing is one of the major requirement. Load is nothing but the of the amount of work that a system performs. Load can be classified as CPU load, memory size and network load. Load balancing is the process of dividing the task among various nodes of a distributed system to improve both resource utilization and job response time. Also avoiding a situation where some of the nodes are heavily loaded and others are idle. Load balancing ensures that every node in the network having equal amount of work (as per their capacity) at any instant of time. In This paper we survey the existing load balancing algorithms for a cloud based environment.

**Keywords** : *Ant colony , honey Bee, Active Clustering ,Biased random Sampling.*

\*\*\*\*\*

## I. INTRODUCTION

When you store your photos online instead of on desktop, or use E-mail or a social networking site, we are using a “cloud computing” services. Cloud computing refers to the delivery of computing resources over the Internet. Instead of keeping data on our own hard drive or updating applications for your needs, we use a service over the Internet, at another location, to store your information or use its applications.

Cloud computing is nothing but providing of computing services over the Internet. Cloud services allow individual person and businesses to use software and hardware that are managed by third parties at remote locations. some Examples of cloud services include online file storage, social networking sites, webmail, and online business applications [5].

## II. MOTIVATION

Due to rapid increase in use of Cloud Computing, moving of more and more applications on cloud and demand of clients for more services and better results, resource management in Cloud has become a very interesting and important research area [4].

Larger companies primarily approach cloud services as a cost-saving strategy for offloading non-mission-critical workloads or those exempt from compliance requirements. Of survey respondents using cloud, 73% point to cost savings as the primary motivating factor. Approximately 65% note cloud computing fits the business' computing needs.

While there are numerous benefits to cloud, 60% of respondents using this cloud model cite improved availability for computing workloads as the biggest benefit. About 57% list workload scalability, which allows users to adjust IT resources to accommodate changes in computing demands, as the main perk for cloud [10].

However, day by day subscribers needs are increasing for computing resources and their needs have dynamic heterogeneity and platform irrelevance. But in cloud computing environment, Load balancing is done and if they are not properly distributed then it will result into resource wastage [8].

Due to the multi-tenant nature, load balancing become a major challenge for the cloud. According to a 2010 survey, it is the second most concerned problem that CTOs express after security. Cloud operators have a variety of distinct resource management objectives to achieve. For example, a public cloud such as Amazon may wish to use a workload consolidation policy to minimize its operating costs, while a private enterprise cloud may wish to adopt a load balancing policy to ensure quality of service [1].

## III. ISSUES IN CLOUD COMPUTING

A cloud computing provides several compelling features and attractive utilities there are some issues in this field which are to be carried out in research. Though there are many issues in Cloud Computing three important issues are discussed [11].

### A. Job Scheduling

Assigning an appropriate number of tasks to the nodes is Job Scheduling. Job scheduling is most important task in cloud computing environment because user have to pay for resources used based upon time. It is one of the major activities executed in many distributed networks, which gives maximum profit to the network. The objective of scheduling is spreading the load among the system equally by maximizing the utilization and minimizing the through-put (task execution time).

### B. Data Security

Huge data are stored in a remote geographical network in a Cloud, so managing the big data is a major issue in Cloud Computing. There is a possibility where malicious user can penetrate into the cloud to hack the data, which infects the entire Cloud.

### C. Load balancing

Load Balancing is a method of distributing workload across multiple computing resources such as cluster of computers, network links. The goal of LB is to optimize the resource usage, evade overload, maximize throughput and to minimize the response time. This was identified as a major concern in Cloud Computing to scale up the increasing demands. It is divided into two types Static Load Balancing and Dynamic Load Balancing.

### D. Data availability

Data should be available at any time anywhere in any place in cloud computing but it is bit complex. We mainly focus on Load balancing issue if cloud computing.

## IV. LOAD BALANCING IN CLOUD COMPUTING

It is a process of reassigning the total load to the individual nodes of the collective system to make resource utilization effective and to improve the response time of the job, simultaneously removing a condition in which some of the nodes are over loaded while some others are under loaded. A load balancing algorithm which is dynamic in nature does not consider the previous state or behavior of the system, that is, it depends on the present behavior of the system [12].

Load Balancing is a technique to distribute the load evenly among all the nodes of the network. If any node is heavy i.e. have more load than required then its load is given to the node with less load. Hence load balancing helps the overloaded and under loaded nodes. Load balancing is a major challenge of cloud computing.

The important things to consider while developing Load balancing algorithm are [11]:

- Estimation of load
- Comparison of load
- Stability of different system
- Performance of system
- Interaction between the nodes
- Nature of work to be transferred
- Selecting of nodes

This load considered can be in terms of CPU load, amount of memory used, delay or Network load.

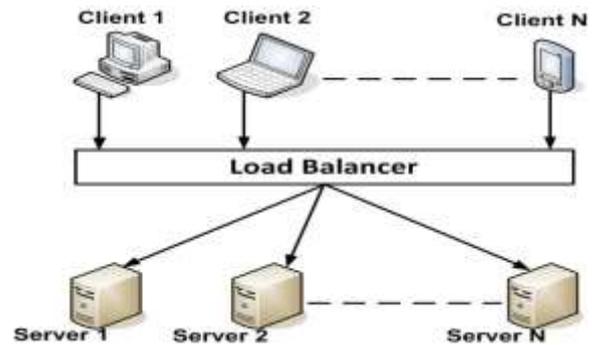


Figure 1: Load Balancer

### A. Goal of Load Balancing

- Improve Performance: it improves the performance by maximizing throughput.
- Maximum Throughput: it achieves the maximum performance and utilization of system resources to avoid ideal nodes in the system resources.
- Fault Tolerance: it ensure the given request will be process by web server.
- Migration Time: it require less migration time.
- Overhead: it minimizes the system overhead cause by load balancing operation.
- Maximum Response Time: it maximize the response time by providing request to the ideal nodes immediately.
- Improve Scalability: by applying load balancing to the cloud , system will be scalable.

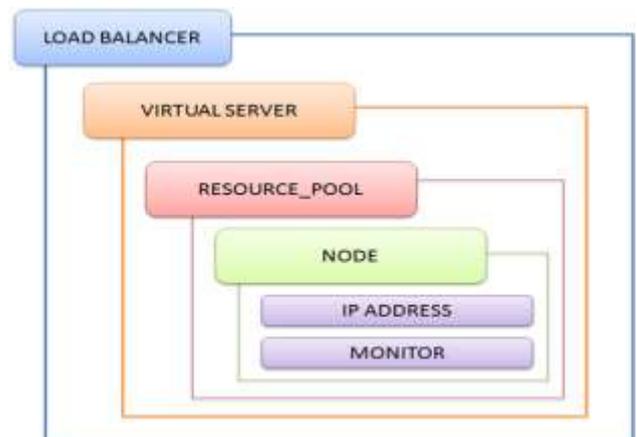


Figure 2: Layer of Load Balancing

### B. Basic Types of Load balancing

The concept of load balancing is mainly depend on the

- Depending on who initiated the process, load balancing can be of three catagories
  1. Sender Initiated: If the load balancing algorithm is initialised by the sender.
  2. Receiver Initiated: If the load balancing algorithm is initiated by the receiver.
  3. Symmetric: It is the combination of both sender initiated and receiver initiated.

- Depending on the current state of the system, load balancing can be divided into two categories.
  - Static
  - Dynamic

enhancing the overall performance of system. Load balancing algorithm should avoid overloading or under loading of any specific node. But, in case of a cloud computing environment the selection of load balancing algorithm is not easy because it involves additional constraints like security, reliability, throughput etc. So, the main goal of a load balancing algorithm in a cloud computing environment is to improve the response time of job by distributing the total load of system. The algorithm must also ensure that it is not overloading any specific node.

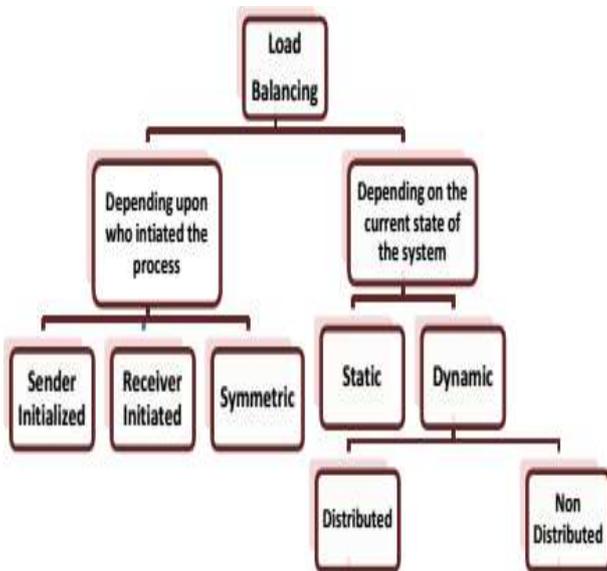


Figure 3: Types of Load Balancing

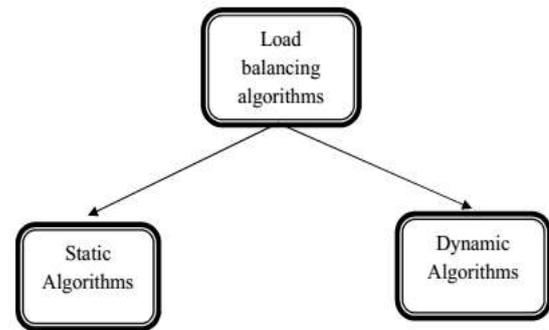


Figure 4: Load Balancing Algorithms Types

C. *Static Load balancing*

In this a load balancing doesn't depend on the current state of the system. Prior knowledge of the system is needed [10]. 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. Although static environment is easier to simulate but is not well suited for heterogeneous cloud environment.

D. *Dynamic Load balancing:*

In this Decisions on load balancing are based on current state of the system. No prior knowledge is needed. So it is better than static approach [10]. In dynamic environment the cloud provider 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. The requirements of the users are granted flexibility (i.e. they may change at run-time). Dynamic environment is difficult to be simulated but is highly adaptable with cloud computing environment.

V. LOAD BALANCING ALGORITHMS

The load balancing can be done by using different algorithms. There are many algorithms which are used for balancing load in cloud computing environment. All the algorithms are mainly divided in basic two categories are static and dynamic [6].

Load balancing is a generic term which is used for distributing a larger processing load to smaller processing nodes for

A. *Static load balancing algorithms :*

- Round-robin
- OLB (opportunistic Load balancing)
- Map Reduce
- Min-Min
- Max-Min

B. *Dynamic load balancing algorithms:*

- Ant Colony optimization
- Honey Bee Foraging
- Biased Random Sampling
- Active Clustering

We mainly focused on Dynamic Load Balancing algorithms.

VI. DYNAMIC LOAD BALANCING

Dynamic load balancing is a major requirement for a successful implementation of cloud environments. The main goal of a cloud-based architecture is to provide elasticity, the Ability to expand and contract capacity on-demand. Sometimes additional instances of an application will be required in order for the architecture to scale and meet demand. That means there is a need for a mechanism to balance requests between two or more instances of that application. The mechanism most likely to be successful in performing such a task is a load balancer [4] [11] [1]. Dynamic Load balancing depends on the current state of the system. If any node is overloaded then its load is shifted to the under loaded node. So real time communication is performed here.

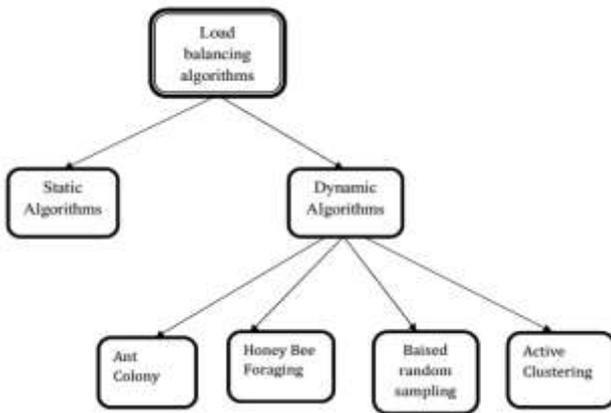


Figure 5: Dynamic Load Balancing

A. Load Balancing Algorithm Based on Ant Colony

Optimization (ACO)

ACO is a probabilistic technique for solving computational problems which can be reduced to finding efficient paths through graphs. The aim of ant colony optimization is to search for an optimal path in a graph on the behavior of ants seeking a path between their colony and source of food. The approach aims at efficient distribution of the load among the nodes and such that the ants never encounter a dead end form movements to nodes for building an optimum solution set.

When request is initialized the ants start their movement by originating from the head node. During the movement, ants deposit a substance called pheromone. The intensity of the pheromone can vary on various factors like the quality of food sources, distance of the food, etc. The ants use these pheromone trails to select the next node. A modified algorithm has been proposed which has an edge over the original approach in which each ant build their own individual result set and it is later on built into a complete solution. However, in this approach the ants continuously update a single result set rather than updating their own result set [7].

Ant based control system was designed to solve the load balancing in cloud environment. Each node in the network was configured with Capacity that accommodates a certain, Probability of being a destination, Pheromone (or probabilistic routing) table. Each row in the pheromone table represents the routing preference for each destination, and each column represents the probability of choosing a neighbour as the next hop. Ants are launched from a node with a random destination. In this approach, incoming ants update the entries the pheromone table of a node. For instance, an ant traveling from (source) to (destination) will update the corresponding entry in the pheromone table in. Consequently, the updated routing information in can only influences the routing ants and calls that have as their destination. In this approach for updating pheromone is only appropriate for routing in symmetric networks.

If an ant is at a choice point when there is no pheromone, it makes a random decision. However, when only pheromone from its own colony is present there is a higher probability that it will choose the path with the higher concentration of its own pheromone type. In addition, due to repulsion, an ant is less likely to prefer paths with (higher concentration of) pheromone from other colonies. Moreover, it is reminded that the degrees of attraction and repulsion are determined by two weighting parameters.

Algorithm

Initialize  
 Node= virtual server  
 Ants= request  
 Destination =web server

AntColonyOptimization ( )

```

{
  Step 1 : Initialize pheromone table
  Step 2: Declare the threshold value for nodes
  Step 3: Ants moves through nodes
    If Load < threshold
      Traverse to node with maximum trailing pheromone
    else
      Traverse to node with minimum foraging pheromone
  Step 4: Update pheromone tables
  Step5: Reassign resources if node is under / overloaded
}
    
```

Calculate pheromone level :

CalculatePheromone ( )

```

{
  Pheromone probability = No. of task having in scheduler of Vin
  Threshold of vi
}
    
```

B. Honeybee Foraging Algorithm

This algorithm is derived from the behavior of honey bees for finding and reaping food. There is a class of bees called the forager bees which forage for food sources, upon finding one, they come back to the beehive to advertise this using a dance called waggle dance. The display of this dance, gives the idea of the quality or quantity of food and also its distance from the beehive. Scout bees then follow the foragers to the location of food and then began to reap it. They then return to the beehive and do a waggle dance, which gives an idea of how much food is left and hence results in more exploitation or abandonment of the food source [4].

In case of load balancing, as the web servers demand increases or decreases, the services are assigned dynamically to regulate the changing demands of the user. The servers are grouped under virtual servers (VS), each VS having its own virtual service queues. Each server processing a request from its queue calculates a profit or reward, which is analogous to the quality that the bees show in their waggle dance. One measure of this reward can be the amount of time that the CPU spends on the processing of a request. The dance floor in case of honey bees is analogous to an advert board here. This board is also used to advertise the profit of the entire colony.

Each of the servers takes the role of either a forager or a scout. The server after processing a request can post their profit on the advert boards with a probability of pr. A server can choose a queue of a VS by a probability of px showing forage/explore behavior, or it can check for advertisements (see dance) and serve it, thus showing scout behavior. A server serving a request, calculates its profit and compare it with the colony profit and then sets its

px. If this profit was high, then the server stays at the current virtual server; posting an advertisement for it by probability pr. If it was low, then the server returns to the forage or scout behavior [4].

Algorithm

Initialization-

si in Vj serving Q,  
 revenue rate interval Tx,  
 Advert:posting prob P,reading prob n,  
 read interval Tr.

Step 1: Forever  
 Step 2: While Q Not Empty do // serving queue  
     Serve request;  
 Step 3: if Tx expired then  
     Compute revenue rate;  
     Adjust n from lookup table;  
 Step 4: if Flip (p) == True then post Advert;  
 Step 5: Tr expired && Read(n) == True then  
 Step 6: if Forager then Select/Read advert id Vk  
     // randomly select  
     Else virtual server id Vk  
     // randomly select  
 Step 7: if Vk Not EQ Vj then Switch(Vk)  
     // migrate to virtual server Vk  
 Step 8: End While  
 Step 9: End Forever

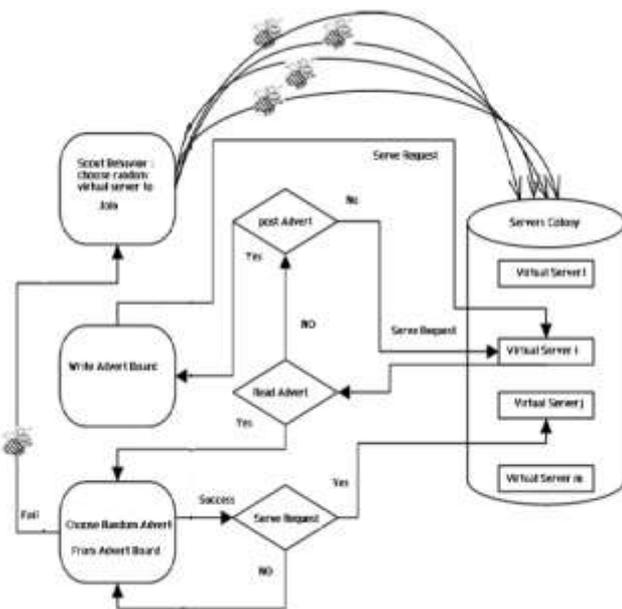


Figure 6: Flow Diagram

The shorter the make span the higher the probability of the solution path. According to this algorithm gives a significant improvement in average execution time and reduction in waiting time of tasks on queue [8].

C. Biased Random Sampling Load Balancing

Here a virtual graph is constructed, with the connectivity of each node (a server is treated as a node) representing the load on the server. Each server is symbolized as a node in the graph, with each indegree directed to the free resources of the server. Regarding job execution and completion [4].

- Whenever a node does or executes a job, it deletes an incoming edge, which indicates reduction in the availability of free resource.
- After completion of a job, the node creates an incoming edge, which indicates an increase in the availability of free resource.

The addition and deletion of processes is done by the process of random sampling. The walk starts at any one node and at every step a neighbor is chosen randomly. The last node is selected for allocation for load. Alternatively, another method can be used for selection of a node for load allocation, that being selecting a node based on certain criteria like computing efficiency, etc. Yet another method can be selecting that node for load allocation which is underloaded i.e. having highest in degree [4].

If b is the walk length, then, as b increases, the efficiency of load allocation increases. We define a threshold value of b, which is generally equal to log n experimentally. A node upon receiving a job, will execute it only if its current walk length is equal to or greater than the threshold value. Else, the walk length of the job under consideration is incremented and another neighbor node is selected randomly. When, a job is executed by a node then in the graph, an incoming edge of that node is deleted. After completion of the job, an edge is created from the node initiating the load allocation process to the node which was executing the job. Finally what we get is a directed graph. The load balancing scheme used here is fullydecentralized, thus making it apt for large network systems like that in a cloud.

Algorithm

BaisedRandomSampling( )

step 1 start  
 Step 2: For each task in task queue  
 Init walklength =0;  
 Step 3: While ( task is assigned to vm ) or ( walklength > threshold)  
     Step 3.1 Increment walklength  
     Step 3.2 Assign task to Vm if indegree > 0  
     Step 3.3 Decrement indegree  
 Step 4 Remove task from task Queue  
 Step 5: End  
 Process completed tasks( )  
 {  
     Increment indegree of vm assigned to the task  
 }

D. Active Clustering

Active Clustering works on the principle of grouping similar nodes together and working on these groups. The process involve [8].

- A node initiates the process and selects another node called the matchmaker node from its neighbors satisfying the criteria that it should be of a different type than the former one.

- The so called matchmaker node then forms a connection between a neighbor of it which is of the same type as the initial node.
- The matchmaker node then detaches the connection between itself and the initial node.

Active clustering is an enhanced method of random sampling, This method uses the resources efficiently thereby increases the throughput and performance of the system by using high resources.

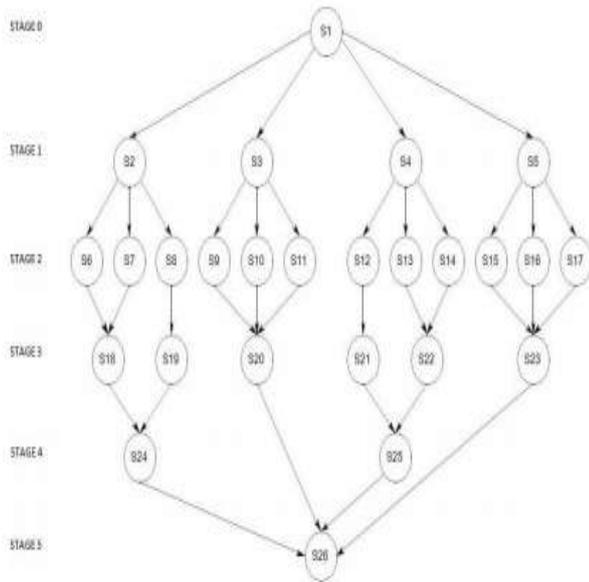


Figure 7: Example

The time required for completing a task within one process is very high. So the task is divided into no. of sub-tasks and each sub-task is given one job. Let the task S is divided into no. of sub-tasks S1, S2,S3...Sn. Out of these some are executed sequentially and some are executed parallelly. So the total time period for completing the task decreases and hence the performance increases. These sub-tasks can be represented in a graph structure known as state diagram.

An example: S1 is executed first. S2, S3, S4 and S5 can be executed parallel during the same time slice. S18 requires the execution of S6 and S7 both, but S19 requires the execution of S8 and so on for all the sub tasks as shown in the state diagram. Our aim is to execute these tasks in different nodes of a distributed network so that the performance can be enhanced.

VII. COMPARISON ANYALYSIS OF DYNAMIC LOAD BALANCING ALGORITHMS

Table 1: Comparison of Dynamic Load Balancing Algorithms

ALGORITHM	PARAMETERS USED	MERITS	DEMERITS	CONCEPT
Honey Bee	-Throughput, -Job completion time -Overhead	-Achive Global Load Balancing, -Maximize Resource utilization, -Low Overhead	-Low priority Load	-Local server action is responsible for Global load balancing
Based random sampling	-Threshold value Or maximum walk length	-Stabilize Load among nodes	-Performance degrade as no. of serve	-Random sampling of the system domain is used to balance the load across all nodes of the system.
Active clustering	-Resourse utilization	-Optimize Job assignment by connecting similar services by local re-wiring	-This degrade the performance when increase in diversity of nodes.	-Self Aggregation algorithm is used to optimize job assignments by connecting similar services by local re wiring
Ant colony	-Fault tolerance, -Resource Utilization, -Scalability	-High fault tolerance -resource utilization, -good scalability	-Complex network and need to manage pheromone table	-Pheromone trail are used to assign the path for the process.

CONCLUSION

The load balancing algorithms were first classified as static and dynamic. Various algorithms under each class along with their variations were also studied. Static Load Balancing algorithms attempt to achieve optimal utilization of resource by considering the size of the tasks and the machines. However, such information may not always be available at hand. Moreover some static load balancing algorithms like the min-min algorithm may cause heavy tasks to starve.

REFERENCES

- [1] Aayush Agarwal, Manisha G, Raje Neha Milind, " a survey of cloud based load balancing techniques" Department of Information Science and Engineering, P.E.S University, 100 Feet ring Road, Banashankari Stage III. Bangalore – 85
- [2] Ruhi Gupta, "Review on Existing Load Balancing Techniques of Cloud Computing." International Journal of Advanced Research inComputer Science and Software Engineering,
- [3] Ratan Mishra , and Anant Jaiswal , " Ant colony Optimization: A Solution of Load balancing in Cloud" International Journal of Web & Semantic Technology .
- [4] Ram Prasad Padhy ,P Goutam Prasad Rao , " load balancing in cloud Computing systems" Department of Computer Science and Engineering National Institute of Technology, Rourkela.
- [5] Shagufta Khan, "Ant Colony Optimization for Effective Load Balancing In Cloud Computing" international

- Journal of Emerging Trends & Technology in Computer Science (IJETTCS)
- [6] J. Uma, V. Ramasamy, A. Kaleeswaran, "Load Balancing Algorithms in Cloud Computing Environment - A Methodical Comparison, International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)
- [7] Ranjan Kumar, "load balancing using ant colony in cloud Computing" International Journal of Information Technology Convergence and Services (IJITCS)
- [8] Pragati Priyadarshinee, Pragya Jain, "Load Balancing and Parallelism in Cloud Computing" International Journal of Engineering and Advanced Technology (IJEAT) ISSN: 2249 – 8958, Volume-1, Issue-5, June 2012
- [9] Shobhana Kashyap , Dr. A.k. Sharma, "load balancing techniques in cloud computing environment" International Conference on Electrical, Electronics & Computer Science Engineering, 26th May-2013, New Delhi, ISBN:978-93-82208-94-594
- [10] Indresh Gangwar, "Juxtaposition of Load Balancing Algorithms in Cloud Computing using Cloud Analyst Simulator" International Journal of Computer Applications (0975 – 8887) Volume 97– No.2, July 2014