Volume 5, Issue 10, October-2015 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Review of Load Balancing Types, Services and Algorithms in Cloud Computing Networks

Haridas Kataria*, Vipul Pant
Lecturer CSE, Govt. Polytechnic for Women,
Sirsa, Haryana, India

***Abstract*— "Cloud computing" is a term, which involves virtualization, distributed computing, networking, software and web services. A cloud consists of several elements such as clients, datacenter and distributed servers. It includes fault tolerance, high availability, scalability, flexibility, reduced overhead for users, reduced cost of ownership, on demand services etc. Central to these issues lies the establishment of an effective load balancing algorithm. The load can be CPU load, memory capacity, delay or network load.**
Our aim is to provide a review of some of the existing methods and algorithms of load balancing in large scale Cloud systems, based on the different performance parameters like throughput, latency etc. for the clouds of different sizes.

***Keywords*— Cloud Computing, Load Balancing, Load Balancing Algorithm**

I. INTRODUCTION

Cloud computing is an emerging computing paradigm which is rapidly gaining consideration in the IT industry. Since cloud computing still is in its infancy, there are many open research challenges. There exist various types of clouds in computing services and these are:

- **Public cloud**: The cloud infrastructure is provisioned for open use by the general public and is made available in a "pay-per-use" manner.
- **Private cloud**: The cloud infrastructure is provisioned for exclusive use by a single organization comprising multiple users.
- **Community cloud**: The cloud infrastructure is provisioned for exclusive use by a specific community of users from organizations that have shared concerns (e.g., mission, security requirements, policy, and compliance considerations).
- **Hybrid cloud**: The cloud infrastructure is a composition of two or more distinct cloud infrastructures (private, community, or public) that remain unique entities, but are bound together by technology that enables data and application portability. A typical example is when a private cloud is temporarily supplemented with computing capacity from public clouds in order to manage peaks in load (also known as "cloud-bursting").

Load Balancing 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.
The focus of the study will be on followings:

- ❖ Types of Load Balancing
- ❖ Types of Load Balancing Algorithms
- ❖ Performance Parameters
- ❖ Strategies in Dynamic Load Balancing

The Section I give the introduction, Section II represents the architecture, Section III represents load balancing types, Section IV focuses on the load balancing algorithms, Section V gives the performance parameters, Section VI compares different algorithms and finally Section VII concludes the work done.

II. ARCHITECTURE

The architecture of a cloud computing system is usually structured as a set of layers. A typical architecture of a cloud system is shown in following figure.
At the lowest level of the hierarchy there is the hardware layer, which is responsible for managing the physical resources of the cloud system, such as servers, storage, network devices, power and cooling systems. On the top of the hardware layer, resides the infrastructure layer, which provides a pool of computing and storage resources by partitioning the physical resources of the hardware layer by means of virtualization technologies. Built on top of the infrastructure layer, the platform layer consists of operating systems and application frameworks.

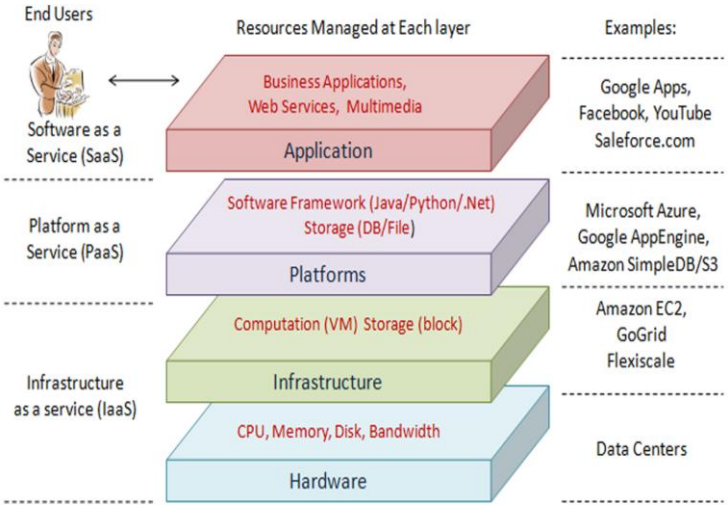

Figure 1: The Architecture of a Cloud System

The purpose of this layer is to minimize the burden of deploying applications directly onto infrastructure resources by providing support for implementing storage, database and business logic of cloud applications. Finally, at the highest level of the hierarchy there is the application layer, which consists of cloud applications.

Service models in Cloud computing are:

• **Infrastructure as a Service (IAAS),** which comprises services to allow its consumers to request computational, storage and communication resources on demand, thus enabling the so called "pay-per-use" paradigm whereby consumers can pay for exactly the amount of resource they use (like for electricity or water). The consumers can use the provided resources to deploy and run arbitrary software; however, the management and control of the underlying cloud infrastructure is possible only by the provider. An example is Amazon EC2.

• **Platform as a Service (PAAS),** which comprises high level services providing an independent platform to manage software infrastructures, where consumers (i.e., developers) can build and deploy particular classes of applications using programming languages, libraries and tools supported by the provider. Usually consumers don't manage or control the underlying infrastructure (such as servers, network, storage, or operating systems), which can only be accessed by means of the high level services provided by the provider. An example is Google App Engine.

• **Software as a Service (SAAS),** which comprises specific end user applications running on a cloud infrastructure. Such applications are delivered to consumer as a network service (accessible from various client devices ranging from desktop computers to smartphones), thus eliminating the need to install and run the application on the consumer's own computers and simplifying maintenance and support. Consumers don't manage or control the underlying infrastructure and application platform; only limited user-specific application configurations are possible. An example is Salesforce.com.

Additional service models have been proposed in literature, but generally they can be regarded as a specialization of the above three service models. For instance, in two extensions are proposed:

(1) An additional model called Hardware as a Service (HAAS), which comprises services for operating, managing and upgrading the hardware.

(2) A specialization of the IAAS model into three categories: -

• **Infrastructure as a Service (IAAS),** which, unlike the above definition, is only concerned to services related to computational resources. An example is Amazon EC2.

• **Data as a Service (DAAS),** which includes services to allow consumers to store their data at remote disks and access them anytime and anywhere. An example is Amazon S3.

• **Communication as a Service (CAAS)**, which provides services related to network communication such as Quality of Service (QoS) management and network security. An example is Microsoft Connected Service Framework.

The traditional approach to deploy a cloud system is a public computing system. However other deployment models are possible which differentiate each other by variations in physical location and distribution.

III. LOAD BALANCING

Based on the approach, the load balancing can be categorized into two parts:

(i) Static Load Balancing: In static load balancing, the performance of the processors is determined at the beginning of execution. Then depending upon their performance the work load is assigned by the master processor. The slave processors calculate their allocated work and submit their result to the master. A task is always executed on the processor to which it is assigned that is static load balancing methods are non pre-emptive. The goal of static load balancing method is to reduce the execution time, minimizing the communication delays.

(ii) Dynamic Load Balancing: Unlike static algorithms, dynamic algorithms allocate processes dynamically when one of the processors becomes under loaded. Instead, they are buffered in the queue on the main host and allocated dynamically upon requests from remote hosts.

A load balancing algorithm which is dynamic in nature does not consider the previous state or behaviour of the system, that is, it depends on the present behaviour of the system. The important things to consider while developing such

algorithm are : 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 etc.

IV. LOAD BALANCING ALGORITHMS

Based on the current state of the system, load balancing algorithms can be classified into two types:

Static algorithm: The current status of the node is not taken into consideration. All the nodes and their properties are known in advance. Based on this prior knowledge, the algorithm works. Since it does not use current system status information, it is easy to implement.

Dynamic algorithm: This type of algorithm is based on the current status of the system. The algorithm works according to the dynamic changes in the state of nodes. Status Table maintains the Current status of all the nodes in the cloud. Dynamic algorithms are complex to implement but it balances the load in effective manner.

Load balancing algorithms can be categorised into three categories based on initiation of process as follows:

1. **Sender Initiated:** In this type the load balancing algorithm is initialized by the sender. Here, the sender sends request messages till it finds a receiver that can accept the load.
2. **Receiver Initiated:** In this type the load balancing algorithm is initiated by the receiver. Here, the receiver sends request messages till it finds a sender that can get the load.
3. **Symmetric:** It is the combination of both the sender initiated and receiver initiated types.

Following load balancing algorithms are currently prevalent in clouds

1. Round-Robin Algorithm [29]

It is the static load balancing algorithm which uses the round robin scheme for allocating job. It selects the first node randomly and then, allocates jobs to all other nodes in a round robin fashion. Without any sort of priority the tasks are assigned to the processors in circular order. Because of the non uniform distribution of workload, this algorithm is not suitable for cloud computing .some nodes get heavily loaded and some nodes get lightly loaded because the running time of any process is not known in advance. This limitation is overcome in the weighted round-robin algorithm .In the weighted round-robin algorithm some specific weight is assigned to the node .on the basis of assignment of weight to the node it would receive appropriate number of requests .If there are equal assignment of weight, each node receive some traffic. This algorithm is not preferred because prior prediction of execution time is not possible

2. Opportunistic Load Balancing Algorithm [30]

This is static load balancing algorithm so it does not consider the current workload of the VM. It attempts to keep each node busy. This algorithm deals quickly with the unexecuted tasks in random order to the currently available node. Each task is assigned to the node randomly. It provides load balance schedule without good results. The task will process in slow in manner because it does not calculate the current execution time of the node.

3. Min-Min Load Balancing Algorithm [31]

The cloud manager identifies the execution and completion time of the unassigned tasks waiting in a queue. This is static load balancing algorithm so the parameters related to the job are known in advance. In this type of algorithm the cloud manager first deals with the jobs having minimum execution time by assigning them to the processors according to the capability of complete the job in specified completion time. The jobs having maximum execution time has to wait for the unspecific period of time. Until all the tasks are assigned in the processor, the assigned tasks are updated in the processors and the task is removed from the waiting queue. This algorithm performs betters when the numbers of jobs having small execution time is more then the jobs having large execution time. The main drawback of the algorithm is that it can lead to starvation.

4. Max-Min Load Balancing Algorithms [31]

Max Min algorithm works same as the Min-Min algorithm except the following: after finding out the minimum execution time, the cloud manager deals with tasks having maximum execution time. The assigned task is removed from the list of the tasks that are to be assigned to the processor and the execution time for all other tasks is updated on that processor. Because of its static approach the requirements are known in advance then the algorithm performed well. It is based on the cases, where meta-tasks contain homogeneous tasks of their completion and execution time, improvement in the efficiency of the algorithm is achieved by increasing the opportunity of concurrent execution of tasks on resources.

5. The two phase scheduling load balancing algorithm [32]

It is the combination of OLB (Opportunistic Load Balancing) and LBMM (Load Balance Min-Min) Scheduling algorithms to utilize better execution efficiency and maintain the load balancing of the system.OLB scheduling algorithm keeps every node in working state to achieve the goal of load balance and LBMM scheduling algorithm is utilized to minimize the execution of time of each task on the node thereby minimizing the overall completion time. This algorithm works to enhance the utilization of resources and enhances the work efficiency.

6. Ant Colony Optimization Based Load Balancing Algorithm [33]

Aim of the ant colony optimization to search an optimal path between the source of food and colony of ant on the basis of their behaviour. This approach aims efficient distribution of work load among the node. When request is initialized the

ant starts movement towards the source of food from the head node. Regional Load Balancing Node (RLBN) is chosen in Cloud Computing Service Provider (CCSP) as a head node. Ants keep records the every node they visits ant record their data for future decision making .Ant deposits the pheromones during their movement for other ants to select next node The intensity of pheromones can vary on the bases of certain factors like distance of food, quality of food etc. When the job gets successful the pheromones is updated. Each ant build their own individual result set and it is later on built into a complete solution. The ant continuously updates a single result set rather than updating their own result set. By the ant pheromones trials, The solution set is continuously updated.

7. Honeybee Foraging load balancing Algorithm [34]

It is a nature inspired decentralized load balancing technique which helps to achieve load balancing across heterogeneous virtual machine of cloud computing environment through local server action and maximize the throughput. The current workload of the VM is calculated then it decides the VM states whether it is over loaded ,under loaded or balanced .according to the current load of VM they are grouped. The priority of the
task is taken into consideration after removed from the overload VM which are waiting for the VM .Then the task is schedule to the lightly loaded VM. The earlier removed task are helpful for the finding the lightly loaded VM. These tasks are known as scout bee in the next step. Honey Bee Behaviour inspired Load Balancing technique reduces the response time of VM and also reduces the waiting time of task.

8. Biased Random Sampling load balancing Algorithm [35]

Biased Random Sampling Load Balancing Algorithm is dynamic approach, the network is represented in the form of virtual graph. Each server is taken as a vertex of the node and the in degree represents the available free resources the nodes have. On the basis of the in degree the load balancer allocates the job to the node. The nodes have at least one in degree then load balancer allocates the job to that node. When the job is allocates to the node then the in degree is incremented again when job gets executed. Random sampling technique is used in the addition and deletion of the processes. The processes are centralized by the threshold value, which indicates the maximum traversal from one node to destination node. The length of traversal is known as walk length. The neighbour node of the current node is selected for the traversal. After receiving the request, load balancer selects a node randomly and compares the current walk length with the threshold value. If the current walk length is equal to or greater than the threshold value, the job is executed at that node. Otherwise, the walk length of the job is incremented and another neighbour node is selected randomly. The performance is decrease as the number of servers increase.

9. Active Clustering load balancing Algorithm [36]

Active Clustering is works on the basis of grouping similar nodes and increase the performance of the algorithm the process of grouping is based on the concept of match maker node. Match maker node forms connection between its neighbours which is like as the initial node .Then the matchmaker node disconnects the connection between itself and the initial node. The above set of processes is repeating again and again. The performance of the system is increases on the basis of high availability of resources, because of that, the throughput is also increasing. This increase in throughput is because of the efficient utilization of resources.

V. PERFORMANCE PARAMETERS

The performance of various load balancing algorithms is measured by the following parameters.

1. Nature: This factor is related with determining the nature or behaviour of load balancing algorithms that is whether the load balancing algorithm is of static or dynamic nature, pre-planned or no planning.

2. Overload Rejection: If Load Balancing is not possible additional overload, rejection measures are needed. Static load balancing algorithms incurs lesser overhead as once tasks are assigned to processors, no redistribution of tasks takes place, so no relocation overhead. Dynamic Load Balancing algorithms incur more overhead relatively as relocation of tasks takes place.

3. Reliability: This factor is related with the reliability of algorithms in case of some machine failure occurs. Static load balancing algorithms are less reliable because no task/process will be transferred to another host in case a machine fails at run-time. Dynamic load balancing algorithms are more reliable as processes can be transferred to other machine in case of failure occurs.

4. Adaptability: This factor is used to check whether the algorithm is adaptive to varying or changing situations .Static load balancing algorithms are not adaptive. Dynamic load balancing algorithms are adaptive towards every situation.

5. Stability: Static load balancing algorithm considered as stable as no information regarding present workload state is passed among processors. However in case of dynamic load balancing such kind of information is exchanged among processors.

6. Predictability: This factor is related with the deterministic or nondeterministic factor that is to predict the outcome of the algorithm. Static load balancing algorithm's behaviour is compile-time. Dynamic load balancing algorithm's behaviour is unpredictable.

7. Forecasting Accuracy: Forecasting is the degree of conformity of calculated results to its actual value that will be generated after execution.

8. Cooperative: This parameter gives that whether processors share information between them in making the process allocation decision other are not during execution. Static algorithms are cooperative and Dynamic algorithms are non cooperative.

9. Resource Utilization: Static load balancing algorithms have lesser resource utilization as static load balancing methods just tries to assign tasks to processors in order to achieve minimize response time ignoring the fact that may be using this task assignment can result into a situation in which some processors finish their work early and sit idle due to lack of work. Dynamic load balancing algorithms have relatively better resource utilization as dynamic load balancing take care of the fact that load should be equally distributed to processors so that no processors should sit idle.

10. Process Migration: Process migration parameter provides when does a system decide to export a process . The algorithm is capable to decide that it should make changes of load distribution during execution of process or not.

11. Preemptiveness: This factor is related with checking the fact that whether load balancing algorithms are inherently non-preemptive as no tasks are relocated. Dynamic load balancing algorithms are both preemptive and non preemptive.

12. Response Time: How much time a distributed system using a particular load balancing algorithm is taking to respond? Static load balancing algorithms have shorter response time. Dynamic load balancing algorithms may have relatively higher response time.

13. Waiting Time: Waiting Time is the sum of the periods spent waiting in the ready queue.

14. Turnaround Time: The interval from the time of submission of a process to the time of completion is the turnaround time.

15. Throughput: Throughput is the amount of data moved successfully from one place to another in a given time period.

16. Processor Thrashing: Processor thrashing occurs when most of the processors of the system are spending most of their time migrating processes without accomplishing any useful work in an attempt to properly schedule the processes for better performance. Static load balancing algorithms are free from Processor thrashing as no relocation of tasks place. Dynamic load balancing algorithms incurs substantial processor thrashing.

VI.. COMPARISON OF ALGORITHMS

As we have discussed in the previous section that there are various parameters available to measure the performance of load balancing algorithms. Among all these parameters, some most important parameters are compared based on their types of algorithms.

	Round Robin	OLB	Min Min	2 phase	Min Max	Ant colony	Honey Bee	Biased Random Sampling	Active Clustering
Throughput	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes	No
Overhead	Yes	No	Yes	Yes	Yes	No	No	No	Yes
Fault tolerance	No	No	No	no	No	No	No	No	No
Response Time	Yes	No	Yes	Yes	Yes	No	No	No	No
Resource Utilization	Yes	Yes	Yes	Yes	Yes	Yes	No	No	Yes
Scalability	No	No	No	No	No	Yes	Yes	No	No
Performance	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No

Figure 2: Comparison of Different Algorithms over Various Parameters

VII. CONCLUSION

Cloud computing provides everything to the user as a service over network. The major issues of cloud computing is Load Balancing. Overloading of a system may lead to poor performance which can make the technology unsuccessful, for the efficient utilization of resources, the efficient load balancing algorithm is required. In this paper, we have surveyed various load balancing algorithms in the Cloud environment. We have discussed the already proposed algorithms by various researchers. The various load balancing algorithms are also being compared here on the basis of different types of parameter

REFERENCES

[1] Anthony T.Velte, Toby J.Velte, Robert Elsenpeter, Cloud Computing A Practical Approach, TATA McGRAW-HILL Edition 2010.

[2] Martin Randles, David Lamb, A. Taleb-Bendiab, A Comparative Study into Distributed Load Balancing Algorithms for Cloud Computing, 2010 IEEE 24th International Conference on Advanced Information Networking and Applications Workshops.

[3] Mladen A. Vouk, Cloud Computing Issues, Research and Implementations, Proceedings of the ITI 2008 30th Int. Conf. on Information Technology Interfaces, 2008, June 23-26.

[4] Ali M. Alakeel, A Guide to Dynamic Load Balancing in Distributed Computer Systems, IJCSNS International Journal of Computer Science and Network Security, VOL.10 No.6, June 2010.

[5] Martin Randles, Enas Odat, David Lamb, Osama Abu- Rahmeh and A. Taleb-Bendiab, "A Comparative Experiment in Distributed Load Balancing", 2009 Second International Conference on Developments in eSystems Engineering.

[6] Peter S. Pacheco, "Parallel Programming with MPI", Morgan Kaufmann Publishers Edition 2008

[7] Mequanint Moges, Thomas G.Robertazzi, "Wireless Sensor Networks: Scheduling for Measurement and Data Reporting", August 31, 2005

[8] Amazon elastic compute cloud (EC2). Available: http://aws.amazon.com/ec2.
[9] IBM Smart Cloud. Available: http://www.ibm.com/cloud-computing.
[10] Linear algebra package (lapack). Available: http://www.netlib.org/lapack/.
[11] Kiam Heong Ang, Gregory Chong, and Yun Li. PID control system analysis, design, and technology. IEEE Transactions on Control Systems Technology,13(4):559–576, 2005.
[12] Panos J. Antsaklis and Anthony N. Michel. Linear Systems. Birhaüser, Boston, 2006.
[13] Danilo Ardagna, Barbara Panicucci, Marco Trubian, and Li Zhang. Energyaware autonomic resource allocation in multi-tier virtualized environments. IEEE Transactions on Services Computing, 99(PrePrints), 2010.
[14] Michael Armbrust, Armando Fox, Rean Griffith, Anthony D. Joseph, Randy H. Katz, Andrew Konwinski, Gunho Lee, David A. Patterson, Ariel Rabkin, Ion Stoica, and MateiZaharia. Above the Clouds: A Berkeley view of Cloud computing. Technical Report UCB/EECS-2009-28, EECS Department, University of California, Berkeley, Feb 2009.
[15] Jr. Arthur E. Bryson and Yu-Chi Ho. Applied Optimal Control: Optimization, Estimation, and Control. Taylor & Francis, revised edition, 1975.
[16] Jerry Banks, John S. Carson, II, Barry L. Nelson, and David M. Nicol. Discrete-Event System Simulation. Prentice Hall, 5th edition, 2010.
[17] Jacques F. Benders. Partitioning procedures for solving mixed-variables programming problems. Numerische Mathematik, 4(1):238–252, 1962.
[18] John R. Birge and François Louveaux. Introduction to Stochastic Programming. Springer Science+Business Media, LLC, 2nd edition, 2011.
[19] S. Bittanti, P. Bolzern, and M. Campi. Exponential convergence of a modified directional forgetting identification algorithm. System Control Letter, 14:131–137, 1990.
[20] Peter Bloomfield. Fourier analysis of time series: An introduction. Wiley- Interscience, 2nd edition, 2000.
[21] Peter Bodík, Rean Griffith, Charles Sutton Armando Fox, Michael Jordan, and David Patterson. Statistical machine learning makes automatic control practical for Internet datacenters. In Proc. of the 2009 USENIX Conf. on Hot Topics in Cloud Computing (HotCloud'09), 2009.
[22] David Booth, Hugo Haas, Francis McCabe, Eric Newcomer, Michael Champion, Chris Ferris, and David Orchard. Web Services Architecture. Working Group Note NOTE-ws-arch-20040211, W3C Web Services Activity, Feb 2004.
[23] George E.P. Box, Gwilym M. Jenkins, and Gregory C. Reinsel. Time Series Analysis: Forecasting and Control. Prentice Hall, 3rd edition, 1994.
[24] Stephen Boyd and Lieven Vandenberghe. Convex Optimization. Cambridge University Press, 2004.
[25] Micheal R. Bussiek and Arne S. Drud. SBB: A new solver for mixed integer nonlinear programming. In Proc. of the 2001 Operation Research Conference (OR'01), 2001.
[26] Rajkumar Buyya, James Broberg, and Andrzej Goscinski, editors. Cloud Computing: Principles and Paradigms. John Wiley & Sons, Inc., Hoboken, NJ, USA, 2011.
[27] Rajkumar Buyya, Rajiv Ranjan, and Rodrigo N. Calheiros. Modeling and simulation of scalable cloud computing environments and the cloudsim toolkit: Challenges and opportunities. In Proc. of the International Conference on High Performance Computing Simulation (HPCS'09), pages 1–11, Leipzig, Germany, 2009.
[28] Rajkumar Buyya, Chee Shin Yeo, Srikumar Venugopal, James Broberg, and Ivona Brandic. Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility. Future Generation Computer Systems, 25(6):599–616, 2009.
[29] Pooja Samal, Pranati Mishra,Round ‖ Robin Algorithms for load balancing in Cloud Computing‖ (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 4 (3) , 2013, 416-419.
[30] Che-Lun Hung1, Hsiao-hsi Wang2 and Yu-Chen Hu2, ―Efficient Load Balancing Alg Network‖. IEEE-78, 2012Vol. 9, pp: 70
[31] T. Kokilavani, Dr. D. I. George Min-Min Algorithm for Static Meta Task Scheduling in Grid computing‖ International Journa-20 No.2, 2011.
[32] Karanpreet Kaur, Ashima Narang, Kuldeep Kaur, "Load Balancing Techniques of Cloud Computing", International Journal of Mathematics and Computer Research, April 2013.
[33] Ratan Mishra and Anant Jaiswal solution of Load Balancing in Web & Semantic Technology (IJWesT), April 2012
[34] Dhinesh B. L.D , P. V. Krishna, ―Honeyinspired bee load balancing of tasks in cloud computing environment proc. Applied Soft Computing, volume 13, Issue 5, May 2013, Pages 2292-2303.
[35] Rahmeh OA, Johnson P, Taleb-Bendiab A.,‖A Biased Random Sampling Scheme for scalable and reliable Grid Networks, The INFOCOMP Science, vol. 7, 1-10
[36] Ram Prasad Padhy ,P Goutam Prasad Rao, Load Balancing in Cloud Computing Systems, National Institute of Technology, Rourkela, India, 2011.

