www.ijecs.in
International Journal Of Engineering And Computer Science ISSN:2319-7242
Volume 6 Issue 6 June 2017, Page No. 21602-21611
Index Copernicus value (2015): 58.10 DOI: 10.18535/ijecs/v6i6.11

Cloud Load Balancing: A Perspective Study

Suman Pandey
sumanuptu@gmail.com

Abstract

Cloud computing has boomed its horizon with large pace as commercial infrastructure in the IT industries meeting the vast requirements of computing resources. There are several issues such as load balancing, virtual machine migration, automated service provisioning, algorithm complexity, etc., demanding to be resolved. Each of these issues needs load balancing to be resolved. This aims for distributing the unwanted dynamic workload between the nodes residing in cloud and this desires of every computing resource must be assigned on proficient and reasonable ground.

Load balancing has become crucial for efficient performance in distributed environments. Cloud computing is an emerging technology demanding more services and better results. Thus load balancing for the cloud is very interesting and important research area. Many algorithms are proposed to provide efficient techniques for assigning the client's requests to available cloud nodes.

This paper studies cloud computing along with research challenges in load balancing. Load balancing has been a major issue for cloud computing environment. Efficient load balancing scheme ensures efficient resource utilization by providing the resources to cloud on-demand of users' basis. By implementing appropriate scheduling criteria load balancing may prioritize users. The aim of this study is to peep in various load balancing algorithms to address its challenges in variety of cloud environment. This study provides a perspective view of the latest approaches in load balancing that will certainly help the future researchers in this field

Keywords: Load Balancing, Cloud Computing, Resource Provisioning, Resource Scheduling, Distributed Computing.

I. INTRODUCTION

Cloud computing is the rapidly growing technology which promotes commercial computing. A cloud is basically a platform offering services from a pool of resources and facilitates the usage of the scalable computing resources such as applications, services, and infrastructure over the network using internet. The cloud computing has changed the paradigm of computing and data from PC's and desktop to large data reserves. Cloud computing dynamically assigns resources to the users at their requested time slots which in-turn optimizes the cost in terms of needed in software and hardware. Thus, cloud computing provide a frame work to access computing resources in on demand way.

By virtue of cloud computing, resources can be assigned and released at the users' request. Thus main focus of cloud computing is on resource allocation and scheduling by using certain algorithms and schemes [1]. These directly affect cloud cost as well as performance. Load balancing is very important feature of cloud computing [2]. At the failure of any service this helps continue of the services by employing provisioning and de-provisioning algorithm.

Thus, Load balancing is a process of distributing the dynamic workload across the nodes in the whole cloud. The scenarios of, when some nodes are heavily loaded while others are idle or doing little are preferably avoided. Overall performance of the system along with resource utilization is efficiently increased. Thus load balancing helps satisfying end users. The other important feature, the scalability of cloud computing is implemented by load balancing.

The organization of paper is as follows: The section II gives brief review of cloud along with discussion of its service model and deployment model. The section III describes need of load balancing algorithms also discusses their classification and ends with description of proposed or implemented load balancing algorithms. The section IV addresses the challenges of these load balancing algorithms and finally concludes in section V with certain idea mentioning the future.

II. CLOUD: A Brief Review

A. Definition

A model which provides a convenient network access if requested to a shared reserves of configurable resources such as servers, networks, storage, and services applications requiring minimal interaction of service provider or management effort [3] is defined as cloud computing. The cloud architectures, its deployment strategies and concerned security belong to this definition. Particularly there are five core elements need to be explained.

1. **On-Demand Self-Service**

On request, at a particular time slot, a consumer can use computing resources such as network storage, CPU time, applications etc. in very easy way and without needing any help of service providers of these resources.

2. **Broad Network Access**

These computing resources are provided over the network using Internet and used by different client applications over variety of platforms such as laptops, mobile phones, laptops and PDAs available to the user.

3. **Resource Pooling**

A provider of cloud service collects or pools computing resources together and serves variety of consumers at their request by dynamically assigning and reassigning different physical and virtual computing resources using either the virtualization model or the multi-tenancy [3]. Economies of scale and specialization are behind setting up such a pool-based computing paradigm. This pool-based model make physical computing resources invisible to consumers which are generally unaware of the location, originalities and formation of these resources such as CPU, database etc.) . For example, consumers are unable to know where their data is stored in the Cloud.

4. **Rapid Elasticity**

To the consumers, resources provisioning seems to be infinite and the consumption can increasingly rise to meet peak requirement at any time. Thus, computing resources are immediate rather than persistent means that there are no commitments or contract to scale up the usage at their need, and release them at the finish to scale down.

5. **Measured Service**

Although, computing resources are pooled and shared by multiple consumers using multi-tenancy but using appropriate methods, the usage of these resources can measured for each consumer individually.

6. **Service Model**

B. Service Models

There are following distinguished service models to classify cloud services:

1. **Software as a Service (SaaS)**

Cloud consumers deploy their applications on a hosting environment and which is made accessible via networks e.g. Internet by application users through various clients interfaces such as PDAs' web browser, etc.. Thus cloud consumers do not have any authority or control over the cloud infrastructure employing multi-tenancy system architecture. To optimize security, speed, availability, maintenance and disaster recovery different cloud consumers' applications are organized in a single logical environment on the SaaS cloud. The SaaS include Google Mail, SalesForce.com, Google Docs and so forth.

2. **Platform as a Service (PaaS)**

This is a development platform which allows cloud consumers to develop cloud services and applications and supports the full Software Development Life Cycle (SDLC) such as Saas directly on the PaaS cloud. Thus, the SaaS hosts completed cloud applications whereas PaaS is used as a development platform hosting both completed and in-progress cloud applications. In addition to hosting environment, programming environment to possess development infrastructure and this also includes configuration management, tools and so forth. Google AppEngine is an example of PaaS.

3. **Infrastructure as a Service (IaaS)**

IT infrastructures such as processing, networks, storage and other main computing resources provided in the IaaS cloud is used by the cloud consumers to develop cloud application. In IaaS cloud consumers basically use virtualization to integrate and decompose physical resources in order to meet growing or shrinking demand of computing resources. The basic technique of virtualization is to create virtual machines which are isolated from both the underlying hardware and other VMs. This strategy varies from the multi-tenancy model, which transforms the application software architecture so that multiple instances of multiple cloud consumers can be run on a single application i.e. the same logic machine. Amazon's EC2 is an example of IaaS.

4. **Data storage as a Service (DaaS)**

Creation of virtualized storage becomes a separate cloud service. The DaaS is defined as a specific IaaS. The aim is to optimize cost in dedicated server, post-delivery services, software license and in-house IT maintenance. The DaaS allows paying for actual

usage rather than total payment the site is licensed for the entire database. DaaS provides table-style abstractions to scale out to store and release large volume of data within a very compressed timeframe often too large, too expensive or too slow for most commercial RDBMS to cope with. Amazon S3, Google BigTable, Apache , and HBase, etc. are some examples of this kind of DaaS.

C. Deployment Model

Most recently, the cloud community has four types of cloud deployment models

1. Private Cloud

The cloud infrastructure operates independently in a single organization, and is managed by the organization or a third party regardless whether it is located on or off the premise. With several aspects a private cloud is setup. Firstly, to optimize the utilization of existing in-house resources a private cloud is created. Securing data and creating trust is also objective behind this. Thirdly, cost of data transfer [4] from local IT infrastructure to a Public cloud is still rather considerable. Fourth, organizations always prefer full control over mission-critical activities residing behind their firewalls. Lastly, private clouds are also built for research and teaching purposes.

2. Community Cloud

There are many organizations which share same cloud infrastructure as well as requirements, values, policies and security concerns. The cloud community cares for democratic equilibrium and economic scalability. A third-party or within one of the organizations in the community hosts the cloud infrastructure.

3. Public Cloud

This is a popular cloud computing deployment model in recent times. A public cloud is used by the general public cloud consumers. The service provider fully owns the public cloud within its own defined set of policies, profits, values, charging and costing model. Many popular public clouds are S3, Amazon EC2, Force.com., and Google AppEngine.

4. Hybrid Cloud

The cloud infrastructure generally consists of two or more types of clouds such as public or private community and they with their unique entities are joined together using standardized or proprietary technology. This technique supports application and data portability such as cloud bursting for load balancing between clouds. Organizations use the hybrid cloud model in order to optimize their computing resources to enhance their competencies while controlling their core activities on-premise using private cloud. Issue of standardization and cloud interoperability is caused by hybrid cloud.

III. Load Balancing in Cloud Computing

The load balancing distributes the load among nodes in cloud environment in situations when some nodes are heavily loaded while some node are with too little load assigned and this has become an important issues in cloud computing [5].

A. Need of Load Balancing

Load Balancing in Cloud Computing is needed due to following:

1. To distribute local workload to all the nodes of cloud in efficient way.
2. To continue the provisioning of services in case the system fails.
3. To increase user satisfaction.
4. To enhance the overall performance of the system.
5. To make response time lesser.
6. To achieve optimized resource utilization.

B. Classification of Localization Algorithms in Cloud Computing

On basis of process initialization, Load Balancing is classified into three classes.

1. Sender Initialized- On the request of client a receiver is assigned to him to receive his workload.
2. Receiver Initialized - On receiving acknowledged request from the receiver the sender shares the workload.
3. Symmetric- This combines both sender and receiver initiated type of load balancing algorithm.

On the basis of the current status of the system, Load Balancing algorithms are grouped into two as described below.

1. Static Load Balancing

The static algorithms [6], use prior knowledge about the system such as storage, processing power, data about user's requirements and desired performance and do not require the information regarding current state of the system. In case of sudden failure of system resources and tasks these algorithms cannot be moved to other node in its execution state to balance the load. Round robin algorithm which divides the data traffic equally among servers belongs to this category. To overcome the problems occurring in round robin algorithm, its modified version is proposed called Weighted Round Robin. The main concept of this is that each server is assigned a weight and the server having the highest weight means more assignment. In equally weighted scenario, servers will receive balanced traffic. This method is generally defined in the designing period of the system.

2. Dynamic Load Balancing

On contrary to static algorithms the dynamic algorithms [7] considers the current state of the system in implementing the load balancing. This overcomes issues of static ones. Being complex in nature, they are able to perform better and are fault tolerant. Certain policies considered in dynamic load balancing algorithms are defined as follows:

1. Transfer Policy- Selecting a job to transfer it from a local node to a remote node.
2. Selection Policy-It specifies the processors involved in the load exchange.
3. Location Policy-Selecting a receiver node to transfer job.
4. Information Policy-Collecting information about the node in the system is referred as information policy and is classified further.
   - i. Load Estimation-In this total amount of workload on a processor is estimated.
   - ii. Process Transfer-It is used to decide which job is to be executed locally or remotely.
   - iii. Priority Assignment- In it priority are assigned to processes for executing them locally and remotely.
   - iv. Migration Limiting- It limits on the maximum number of times a task can be migrated from one machine to another machine.

On decision strategy, the load balancing algorithms are categories into three as follows.

1. Centralized Load Balancing- In this category a single node termed as central does all the allocation and scheduling. This node keeps knowledge of entire cloud network and does load balancing in either static or dynamic approach. This way it requires lesser the time to analyse different cloud resources but its flip side is that the centralized node is heavily loaded. This type of network is not fault tolerant as recovery is very difficult in case of failure of centralized node.

2. Distributed Load Balancing- In this, resource allocation or task scheduling are not done by a single node but multiple domains participate in load balancing. In this approach every node in the network maintains a local information base which is used to distribute tasks in static as well as in dynamic environment. In case of failure of a node it continues and makes the system fault tolerant.

3. Hierarchical Load Balancing- This algorithm involves various levels of cloud. In hierarchical load balancing, slave mode operation method is adopted. Tree data structure is used to represent layered structure of clouds and every node in the tree is made balanced under supervision of its parent node. Master node uses agent process of light weight to collect knowledge of slave nodes and parent node make decisions based on gathered information.

IV. Load Balancing Algorithms in Cloud Computing

The load balancing algorithms which are proposed in literature or being implemented belong to either static or dynamic domains.

A. Static Load Balancing Algorithms

1. Round Robin Load Balancing

In the round robin mechanism [8] a time slot is allotted to each process to be executed and it follows in a ring manner and in addition, a balance technique is followed in order to balance the process in a group till all processes completes their task. The processes are executed in round robin fashion till the all processes complete their task. This algorithm is widely implemented in web servers where http requests are of similar nature and distributed equally.

2. Shortest Job Scheduling Algorithm

The shortest job is selected first in this algorithm [9]. This approach completes execution of shortest jobs first to utilize the resources completely for longer jobs. Shortest job has advantage of having shorter waiting time.

3. Min-Min Load Balancing

In this approach all the information related to the job is available in prior to its implementation. Min-Min algorithm [10] starts with a set of jobs waiting in a queue. Firstly, time required to complete a task is estimated and then job with minimum execution time is selected for execution. The cloud node having the minimum execution time for all jobs is chosen and finally, the selected job and the selected node are mapped. The ready time of the node is updated and this process is repeated until all jobs are executed. The job with the shortest execution time is executed first and it may happen that some jobs experience starvation.

4. Max-Min Load Balancing

It works on contrary to min-min approach. The Max-Min [11] performs in same way as the min-min algorithm where the machine that has the minimum completion time for all the jobs is selected and then instead of selecting job with maximum time is selected. Finally the selected job and the selected node are mapped. Then the ready time of the node is updated by adding the execution time of the assigned task.

5. Two-Phase (OLB + LBMM) Load Balancing Algorithm

To achieve better executing efficiency, the author in [12] has merged Opportunistic Load Balancing (OLB) and Load Balance MinMin (LBMM) scheduling algorithms. The OLB algorithm puts each and every node in working condition so as to achieve goal of cloud computing whereas LBMM scheduling algorithm is used to minimize the execution time of the tasks on node which reduces overall completion time. Combining these two algorithms help achieve efficient utilization of all resources and improves performance efficiency of the network of multiple processors as whole.

6. Central Load Balancing Policy for Virtual Machines (CLBVM)

A policy has been proposed by A. Bhadani et al. [13] that balance the load evenly in a cloud computing environment and distributed virtual machine and termed as Central Load Balancing Policy for Virtual Machines (CLBVM). This also improves the overall performance of the system

B. Dynamic Load Balancing Algorithms

1. Power Aware Load Balancing (PALB)

In this [14] approach firstly utilization percentage of each computing node is calculated for the working module and also decides number of computing nodes which are in working condition or in idle condition. There are three working modules named as balance section, upscale section and downscale section in this algorithm. Balance section initializes the process where virtual machine is going to start. The upscale section starts the additional computing nodes and shut-downs non-working computing nodes in participating nodes. The algorithm gives it best performance on consumption as compared to other algorithms in same category.

2. Fuzzy Active Monitoring Load Balancing (FAMLB)

A Load Balancing algorithm is proposed by Srinivas Seth et al. [15] based on fuzzy logic. The processor speed and load on virtual machine are two main parameters of this algorithm. The authors in [16] have introduced a dynamic load balancing algorithm known as fuzzy Active Monitoring Load Balancer (FAMLB) in which bandwidth usage, memory usage, disk space usage and virtual machine status are additional parameters considered.

3. Throttled Load Balancing

The author in [17] has proposed an algorithm where client initiates the process by requesting the load balancer to find a suitable virtual machine to process the incoming tasks. There are multiple instances of virtual machine working in Cloud Computing environment handling various types of requests. As per client's request, the load balancer searches to find a group ready to process the request to be assigned it.

4. Honeybee Foraging Behaviour

A load balancing algorithm proposed in [18] performs similar to honey bees in finding and reaps their food. The forager bees search for food and after getting it they make announcements by dancing called waggle dance. With the information the scout bees follows the searcher bees towards the location to storage food and finally returns to beehive. They again perform waggle dance to inform the availability of food to consume more. In dynamic load balancing, the services are assigned dynamically as per users' changing demands. The virtual servers are grouped to form cluster wherein each virtual server has its own virtual service queue. In same way as bee does waggle dance, each server also estimates a profit or reward corresponding to amount of time that the CPU spends on the processing of a request. This mechanism in Load Balancing as well as in virtual server is also useful while occupying the server.

5. Active Clustering

This load balancing algorithm forms various groups of similar jobs and then does group wise execution, thus enhances the performance. This algorithm performs poorly in increasing diversity in system as described in [19]. A node initiates the process of selecting a node called matchmaker node from its neighbours based on certain criteria. The processes of a group are executed one by one till all processes get executed. The match maker algorithm makes connection between matchmaker node and its neighbour similar to initial node. After processing, matchmaker node disconnects itself from the initial node.

6. Biased Random Sampling

The author M. Randles et al. [20] has proposed scheme which dynamically on random basis samples system data to acquire self-organization and this way it balances the load in all system of node. Here a virtual graph is made showing the connectivity of those nodes showing load on the server regarding job execution and completion time in the network. Whenever a node completes a job it deletes an incoming edge also frees the resources. According to information from random samples the jobs are added or deleted. The process tarts at any one node and in each step a neighbour is selected randomly and the node added in last is selected for assigning the load. Alternatively, a node for load allocation, is selected based on certain criteria such as computing efficiency etc.. The other way is to select a node for load allocation which is under loaded. The load balancing scheme performs in fully decentralized manner and thus making it appropriate for large network systems like in a cloud.

7. Generalized Priority Algorithm

In this approach [21] the tasks are assigned priority based on size of the tasks in a way that the task of highest size gets the highest priority and executed. Virtual servers are also prioritized based on their million instruction per second (MIPS) value and the server with highest MIPS value gets highest priority and thus it balance load optimize utilization of the resources.

8. Join-Idle-Queue

An algorithm proposed by Y. Lua et al. [22] to use web services and systems are termed as Join-Idle-Queue load balancing algorithm. It does large scale load balancing using distributed dispatchers. In each dispatch firstly, load balancing algorithm frees the processors to perform and then perform allotments of the task to processors in such a way reducing the queue length at each server. This algorithm avoids critical path formation in the load balancing work and helps effective reduction of the system load.

C. Genetic Algorithm Based Load Balancing

K. Dasgupta and B. Mandal in [23] have proposed an idea to balance the load of the cloud infrastructure while minimizing span of a job. To distribute load effectively in system genetic based approaches follows some rules and optimization.

1. Ant Colony Optimization

The authors in [24] have proposed meta data heuristic approach for load balancing in cloud. The functioning of this algorithm is similar in nature of real ants which form a network to process its job. Ant colony optimization is based on heuristic which guarantees for optimal balancing a system with any number of machines having any number of jobs. The ants move to search food from source to their nest in a path. The ants form connection with each by dropping a liquid named as pheromone and other ants follow the same path with dropped pheromone. Ants follow that path with the highest intensity of pheromone otherwise no optimal solution is guaranteed. The social agents like birds, ants and honey bees follows optimal path for solving their problems which changes dynamically with changed environments. In the proposal ants are moved on graph where all the nodes are connected and randomly move until an optimal solution is found.

2. Stochastic Hill Climbing Technique

The authors in [25] proposed a novel load balancing mechanism using Stochastic Hill Climbing technique. The hill climbing forms the uphill chosen randomly and moves with a favouring probability. The Stochastic Hill climbing uses the resources for allocating jobs to the servers or virtual machines (VMs).

D. Decentralized Content Aware Load Balancing

A load balancing policy proposed by H. Mehta et al. [26] defines the usage the unique and special property (USP) of the computing nodes and requests also. The USP helps the task scheduler to take the best and fit resources to execute first. With lesser overhead, this method is implemented in decentralized manner and also improves the searching performance by knowing the content information. Thus optimizes the utilization of resources by reducing the idle time of the computing nodes.

1. Server-Load Balancing For Internet Distributed Services

The authors in [27] have proposed a distributed server based load balancing mechanism for web servers. It improves the performance by reducing service latency through implementing a method which binds requests to the closest remote servers

without overloading them. A middleware is needed to implement the methodology. To balance the overload web server uses a heuristic scheme that guarantees the balancing of load based on the size of job and also ensures not to repeat same size job in single node.

2. Load Balancing Based on a Lock Free Multiprocessing Solution

A proposal in [28] avoids the usage of shared memory while other multiprocessing load balancing algorithms use concept of shared memory by locking it for a session and so named as a lock-free multiprocessing solution. This solution helps improves the performance of load balancer in a multi-core environment.

3. Load Balancing Strategy for Virtual Machine Resources

In [29] author has implemented a scheduling strategy using previous logs and the current status of the server. By using a genetic algorithm approach this strategy helps in reduction of dynamic migration in the system. It helps in resolving the issues of load-imbalance and high cost of migration thus achieving better resource utilization. A drawback for the system is that, sometime previous logs cannot give the current scenario at its level best.

4. Load Balancing Strategy for Virtual Storage (LBVS)

H. Liu et al. [30] has proposed a load balancing algorithm providing a large scale net data storage as service (daas) of cloud. Storage virtualization is achieved using three-tier architecture and achieves load balancing by using two load balancing modules. Replica balancing facilitates concurrent access by reducing the response time and enhancing storing capacity. The proposed algorithm increases robustness and flexibility.

5. Load Balancing Based on a Task Scheduling Algorithm

A two-tier task scheduling strategy is proposed in [31]. It meets the changing demands of clients and optimizes resource utilization using Load Balancing. Firstly, tasks are mapped to virtual machines and then virtual machine to host resources which improves resource utilization and task response time also.

6. Load Balancing Based on Ant Colony and Complex Network Theory (ACCLB)

A load balancing mechanism proposed by Z. Zhang et al. [32] which provides small-world and scale-free characteristics of a complex network. This is fault tolerant as well as having good scalability as it overcomes heterogeneity by adapting the dynamic environment. Thus system performance may improve.

7. Event-Driven Mechanism

The author in [33] has proposed an algorithm implementing event driven load balancing. This algorithm uses capacity based mechanism for input capacity processing. On reception of input the algorithm analyses the resources and its components on the basis of global states of the game session and after completion of this session, algorithm generates the game session load balancing action. Event-driven algorithm can scale up or down a game session on multiple resources depending on variable user load. The only drawback is its occasional QoS breaching.

8. CARTON

R. Stanojevic et al. [34] has proposed a mechanism called CARTON to control cloud. The cloud control merges load balancing (LB) and distributed rate limiting (DRL). Load balancing is useful if it equally distributes jobs among servers to minimize the associated costs and DRL is used to ensure fair resource allocation among servers. DRL also adapts server capacities for the dynamic workloads so that performance equality is achieved at each server with very low computation and communication overhead. This algorithm is also simple and easy to implement.

9. Compare and Balance

This algorithm [35] uses the concept of compare and balance to reach the in equilibrium situation and manages balancing of load of systems. A comparison based on the sorting approach is applied. The sorting is based on probability of number of virtual machines running on the current host and whole cloud system. The current node selects randomly a node and compares the load with itself. The algorithm is simple and follow traditional approach for load balancing in cloud computing.

10. Vector Dot Proposal

A Vector Dot dynamic algorithm is proposed by A. Singh et al. [36] for Load Balancing. This algorithm supports the layered structure of data centre and also handles multidimensionality of network switches, resource load across servers and storage in an

agile data centre. This agile data centre implements storage virtualization technologies as well as integrated servers. The dot product is used to distinguish nodes on the basis of item requirements and avoids overloading on servers, switches and storage nodes.

IV. Challenges of Load Balancing in Clod Computing

With a vast and extensive discussion on various load balancing algorithms proposed for cloud. It is found that research particularly in cloud load balancing is still in developing stage and some scientific challenges remain unsolved some of them are discussed below.

1. Automated Service Provisioning

Elasticity is important feature of cloud computing in which resources are dynamically assigned or released to cloud node as per demand of clients automatically. This is very important issue for implementing load balancing and demands for much more advanced techniques.

2. Virtual Machines Migration

Virtualization, a very important feature of cloud where in, the whole machine seems to be a set of files and to unload a heavily loaded machine, its files (load) are moved from one virtual machine to another physical machines. This distributes the load in a data-centre or set of data-centres in cloud. Thus, dynamically distributing the load during the execution state may cause bottlenecks in Cloud computing systems so must be efficiently implemented.

3. Energy Management

Widely adopted technology of cloud computing is for the economy of scale. Energy saving is a challenging issue which advocates global economy. Load balancing algorithms should use resources efficiently to optimize energy usage and complex algorithm should be avoided and is a challenge to the algorithm designer.

4. Stored Data Management

In previous years data storage across the network has exponentially boomed. With the emergence of cloud technology, the commercial sector private or public is demanding (daas) data storage as service to store data for their individuals and the management. Thus management of data storage becomes a major challenge for cloud computing. Distributing data to the cloud for optimizing computing storage while managing fast access is the todays challenge.

5. Emergence of Small Data Centres for Cloud Computing

Small data centres are cheaper, beneficial consume lesser energy to large data centre. Small service providers provide cloud services leading towards geo-diversity computing. To ensure reasonable response time with optimal distribution of resources is a challenge to scientific community.

V. CONCLUSION AND FUTURE SCOPE

In this paper, emergence of cloud computing with its definition of service model and development model were studied. Then after discussing requirements of load balancing various load balancing algorithms proposed or implemented in cloud computing have been studied. Finally the challenges of the algorithms were discussed and concluded with the idea that more efficient load balancing algorithms need to be developed meeting the future demands.

VI. REFERENCES

[1] T. Ma, Y. Chu, L. Zhao and O. Ankhbayar, “Resource Allocation and Scheduling in Cloud Computing: Policy and Algorithm”, IETE Technical Review Volume 31, Issue 1, pp.4-16,January 2014.

[2] B. P. Rima, E. Choi, and I. Lumb, "A Taxonomy and Survey of Cloud Computing Systems", Proceedings of 5th IEEE International Joint Conference on INC, IMS and IDC, Seoul, Korea, pp.44-5l, August 2009.

[3] P. Mell and T. Grance, "Draft NIST Working Definition of Cloud Computing - v15", 21 Aug 2009.

[4] M. Armbrust, A. Fox, R. Griffith, A. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, and I. Stoica, "Above the Clouds: A 32 Berkeley View of Cloud Computing," EECS Department, University of California, Berkeley, Tech. Rep. UCB/EECS-2009-28, 2009.

[5] M. A. Vouk, “Cloud Computing – Issues, Research and Implementations”, Proceedings of the ITI 2008 30th Int. Conf. on Information Technology Interfaces, Cavtat, Croatia June 23-26, 2008,

[6] N. Jain, Kansal and I. Chana,"Cloud Load Balancing Techniques: A Step Towards Green Computing", IJCSI, Vol. 9, Issue 1, January 2012.

[7] H.Bheda and H. Bhatt, "An Overview of Load Balancing Techniques in Cloud Computing Environments", International Journal of Engineering and Computer Science Volume 4, pp.9874- 9881, JANUARY 2015.

[8] N.Pasha, A. Agarwal and R. Rastogi, "Round Robin Approach for VM Load Balancing Algorithm in Cloud Computing Environment" International Journal of Advanced Research in Computer Science and Software Engineering Volume 4, Issue 5, May 2014.

[9] P. Devi and Trilok, "Implementation of Cloud Computing by using Short Job Scheduling" International Journal of Advanced Research in Computer Science and Software Engineering.

[10] H. Chen, F.Wang, N. Helian and G.Akanmu, "User-Priority Guided Min-Min Scheduling Algorithm For Load Balancing in Cloud Computing", Parallel Computing Technologies, National Conference, 2013.

[11] S. Aslanzadeh, V. Mahadevan, and C. Mcdermid, " Availability and Load Balancing in Cloud Computing", International Conference on Computer and Software Modeling IPCSIT vol.14 IACSIT Press, Singapore 2011.

[12] S. Wang, K. Van, W. Liao, and S. Wang, "Towards a Load Balancing in a Three-level Cloud Computing Network", Proceedings of the 3rd IEEE International Conference on Computer Science and Information Technology (ICC SIT), Chengdu,China,pp.108-113,September,2010.

[13] A. Bhadani and S. Chaudhary, " Performance Evaluation of Web Servers using Central Load Balancing Policy over Virtual Machine on Cloud", proceedings of third Annual ACM.

[14] J. M. Galloway, K. L. Smith, and S. S. Vrbsky, "Power Aware Load Balancing for Cloud Computing," In Proceedings of the World Congress on Engineering and Computer Science, vol. 1, pp.19–21, 2011.

[15] S. Sethi, A. Sahu, and S. K. Jena, "Efficient Load Balancing in Cloud Computing using Fuzzy Logic," IOSRJournal of Engineering, vol. 2, no. 7, pp.65–71, 2012.

[16] Z. Nine, M. SQ, M. Azad, A. Kalam, S. Abdullah and R. M. Rahman, "Fuzzy Logic Based Dynamic Load Balancing in Virtualized Data Centers" In fuzzy system (FUZZ), IEEE International conference on, pp. 1-7, 2013.

[17] Ms.Nitika, Ms.Shaveta, and G. Raj, "Comparative Analysis of Load Balancing Algorithms in Cloud Computing", International Journal of Advanced Research in Computer Engineering & Technology Volume 1, Issue 3, May 2012.

[18] M. Randles, D. Lamb, and A. Taleb-Bendiab, "Experiments with Honeybee Foraging Inspired Load Balancing" Proceedings IEEE International Conference on Developments in eSystems Engineering (DESE), pp.240 – 247,Abu Dhabi, Dec 2009.

[19] http://www .loadbalancing.org/.

[20] M. Randles, D. Lamb, and A. Taleb-Bendiab, "A Comparative Study into Distributed Load Balancing Algorithms for Cloud Computing", Proceedings IEEE International Conference on Advanced Information Networking and Applications Workshops, Perth, Australia, pp.551-556, April 2010.

[21]A. Agarwal, and S. Jain "Efficient Optimal Algorithm of Task Scheduling in Cloud Computing Environment", International Journal of computer Trends and Technology (IJCTT), V9(7):344-349, March 2014. ISSN:2231-2803.

[22]G. Kliotb, Y. Lua, Q. Xiea, A. Gellerb, J. R. Larusb, and A. Greenber, "Join-Idle-Queue: A Novel Load Balancing Algorithm for Dynamically Scalable Web Services", An international Journal on computer Performance and evaluation, In Press, Accepted Manuscript, Available online 3 August 201l.

[23] K. Dasgupta, B. Mandal, P. Dutta, J. K. Mandal, and S. Dam , "A Genetic Algorithm (GA) based Load Balancing Strategy for Cloud Computing", Elsevier (CIMTA), 2013.
[24] A. Jain, and R. Singh, " Review of Peer to Peer Grid Load Balancing Model Based on Ant Colony Optimization with Resource Management" Volume 3, Issue 4, IJARCSSE, April 2013.

[25] K. Dasgupta, B. Mandal, and P. Dutta, "Load Balancing in Cloud Computing using Stochastic Hill Climbing-A Soft Computing Approach", Elsevier (C3IT) 2012.

[26] H. Mehta, P. Kanungo, and M. Chandwani, "Decentralized content aware load balancing algorithm for distributed computing environments", Proceedings of the International Conference Workshop on Emerging Trends in Technology (lCWET),pp.370-375, February 2011.

[27] A. M. Nakai, E. Madeira, and L. E. Buzato,"Load Balancing for Internet Distributed Services Using Limited Redirection Rates", 5th IEEE Latin-American Symposium on Dependable Computing (LADC),pp.156-165 2011.

[28] Xi. Liu, Lei. Pan, Chong-Jun. Wang, and JunYuan. Xie, "A Lock-Free Solution for Load Balancing in Multi-Core Environment", 3rd IEEE International Workshop on Intelligent Systems and Applications (lSA), pp.1-4 2011.

[29] Hu, 1. 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.

[30] H. Liu, S. Liu, X. Meng, C. Yang, and Y. Zhang, "LBVS:A Load Balancing Strategy for Virtual Storage', IEEE International Conference on Service Sciences, 2010.

[31] Y. Fang, F. Wang, and J. Ge, "A Task Scheduling Algorithm Based on Load Balancing in Cloud Computing", Web Information Systems and Mining, Lecture Notes in Computer Science, Vol. 6318,pp.271-277,2010.

[32] Z. Zhang, and X. Zhang, "A Load Balancing Mechanism Based on Ant Colony and Complex Network Theory in Open Cloud Computing Federation", Proceedings of 2nd International Conference on Industrial Mechatronics and Automation (ICIMA), Wuhan, China,pp.240- 243, May 2010.

[33] V. Nae, R. Prod an, and T. Fahringer, "Cost Efficient Hosting and Load Balancing of Massively Multiplayer Online Games", Proceedings of the fifth IEEE/ ACM International Conference on Grid Computing (Grid), IEEE Computer Society,pp.9- 17, October 2010.

[34] R. Stanojevic, and R. Shorten, "Load Balancing vs. Distributed Rate Limiting: A Unifying Framework for Cloud Control", Proceedings of IEEE ICC, Dresden, Germany, pp. 1-6, August 2009.

[35] Y. Zhao, and W. Huang, "Adaptive Distributed Load Balancing Algorithm based on Live Migration of Virtual Machines in Cloud", Proceedings of 5th IEEE International Joint Conference on INC, IMS and IDC, Seoul, Republic of Korea,pp.170-175, August 2009.

[36] A. Singh, M. Korupolu, and D. Mohapatra, "Server-Storage Virtualization: Integration and Load Balancing in Data Centers", Proceedings of the ACM/IEEE conference on Supercomputing (SC), November 2008.