LOAD BALANCING IN CLOUD COMPUTING: A SURVEY

Nagamani H. Shahapure[1] and P Jayarekha[2]
[1]Department of Information Science and Engineering, VTU, Bangalore, Karnataka, India
[2]Department of Information Science and Engineering, BMSCE, Bangalore, Karnataka, India

ABSTRACT
Cloud computing is a computing paradigm in which dynamically scalable virtualized computing resources are provided as a service over the Internet. It can be viewed as a collection of computing and communication resources shared by active users. Scalability is a desirable attribute of a network, system, or process. Scalability refers to the ability to service large number of users. The increase in the demand for computing power increases the need for scalability of the system. This results in the fulfilment of the ever increasing user demands. Poor scalability results in poor performance of the system. An important factor that is central to scalability is load balancing. Load balancing is one of the main challenges in cloud computing. This is required to distribute the dynamic workload across multiple nodes to ensure that no single node is overwhelmed. These techniques should minimize execution speeds and communication delay and maximize resource utilization and throughput. Research in this area has led into the development of algorithms called as load balancing algorithms. In this survey paper we present the performance analysis of various load balancing algorithms based on different parameters like efficient scheduling, in-time job completion. The two commonly used load balancing algorithms static and dynamic are discussed. The analysis indicates that both types of algorithms have their own advantages and disadvantages. Deciding the type of algorithm to be implemented to obtain scalability will be based on type of applications to solve. The main purpose of this paper is to help in design of new algorithms by studying the behaviour of various existing algorithms.

KEYWORDS: *Cloud Computing, Scalability, Load Balancing, Static, Dynamic.*

I. INTRODUCTION

Cloud Computing is a computing model, where resources such as computing power, storage, network and software are abstracted and provided as services in a distributed environment. Billing models for these services are based on the ones adopted for public utilities. Data is ubiquitous and is distributed over a heterogeneous environment. In simple words it can be defined as a technology where the work is executed by sharing and using resources and applications of a network environment. This is done without being concerned about the owner and manager of these resources. [7]
Applications that run on cloud operate at large scale similar to internet. This requires handling more number of simultaneous requests than any enterprise application. The different types of services provided by cloud systems are software services (software as a service, SaaS) or physical services (platform as a service, PaaS) or hardware/infrastructure service (Infrastructure as a Service, IaaS).
The objective of the paper is to understand the load balancing techniques in cloud. Cloud and cloud components are discussed briefly in section II. Section III gives a brief overview of scalability. Section IV explains about load balancing. Section V explains about the parameters for load balancing. Section VI and VII gives an overview of the existing load balancing algorithms. Section VIII concludes the paper with directions for future work.

II. CLOUD AND CLOUD COMPONENTS

Cloud computing is a distributed computing paradigm that focuses on providing a wide range of services to different users. The users can get access to scalable, virtualized hardware and/or software infrastructure over the network. Cloud computing involves distributed technologies to satisfy a variety of applications and user needs. [17]
To be more specific, a cloud is a platform or infrastructure that enables execution of code (services, applications etc.), in a managed and scalable fashion. The cloud is well managed indicates that the requests of the users are satisfied in terms of quality and time. Scalable implies that the resources are put to use according to actual current requirements.

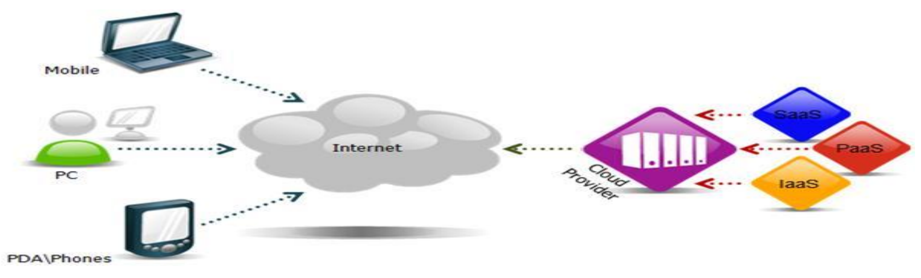

Figure 1 Cloud System

1.1. Cloud Components

A cloud system has 3 major components. These components are shown in figure 2.
1.1.1. Clients: End users interact with the clients to manage information related to the cloud. Clients can be a browser, hand held device.
1.1.2. Data Center: It is a collection of servers hosting different applications. An end user connects to the data center to subscribe different applications.
1.1.2. Distributed Servers: These are parts of a cloud which are present in the internet for hosting different applications.

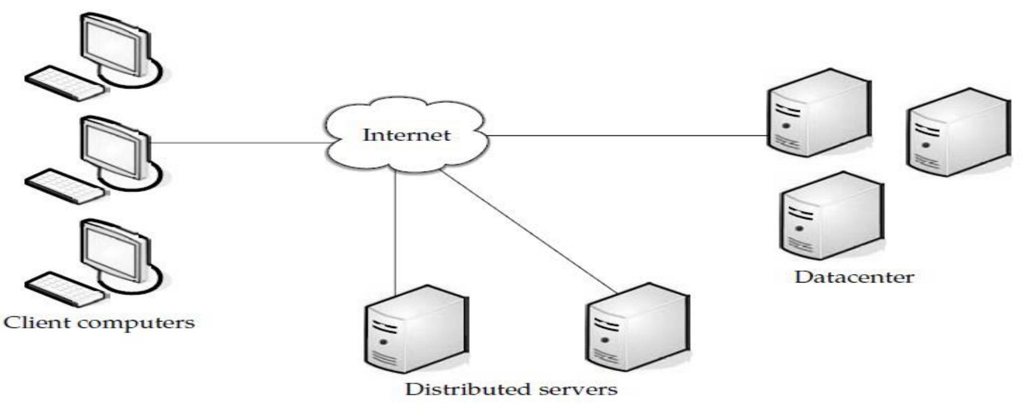

Figure 2 Components of a cloud

III. SCALABILITY

An important challenge in the cloud is the issue of scalability. It is an important parameter in a network system or a process. The ability of a system to accommodate an increasing number of elements or objects, to process growing volumes of work without any delay is scalability. When procuring or designing a system, we often require that it be scalable. A system, whose performance

improves after adding hardware, proportionally to the capacity added, is said to be a scalable system. Similarly, an algorithm is said to scale if it is suitably efficient and practical when applied to large situations (e.g. a large input data set or large number of participating nodes in the case of a distributed system). If the algorithm fails to perform when the resources increase then it does not scale.
One of the important issue is scalability is the load scalability. It deals with the system's capacity to handle the ever increasing user demands. A scalable algorithm must be efficient in handling multiple user requests without any interruption. Techniques have been developed to handle the ever increasing load in the cloud. These algorithms achieve scalability through load balancing.

IV. LOAD BALANCING

Load balancing is a technique used to distribute the load over the different nodes which provides good resource utilization. It also removes the condition where some of the nods are overloaded and some others are under loaded. When a particular node is crashed or overloaded with the data then load is distributed over the other ideal nodes. The important things to consider while designing a load balancing algorithm are: load estimation, load comparison, stability of different system, performance of system, interaction between the nodes, nature of work to be transferred, selecting of nodes and many other ones. This load can be in terms of CPU load, amount of memory used, delay or network load. [18]
In general terms load balancing can be defined as the distribution of processing and communication activities evenly across a network. This is done in order to avoid overwhelming of a single device. Load balancing is especially important for networks where it's difficult to predict the number of requests that will be issued to a server. Busy web sites typically employ two or more web servers in a load balancing scheme. If one server starts to get flooded, requests are forwarded to another server with more capacity. Dividing the traffic between servers, data can be sent and received without major delay.

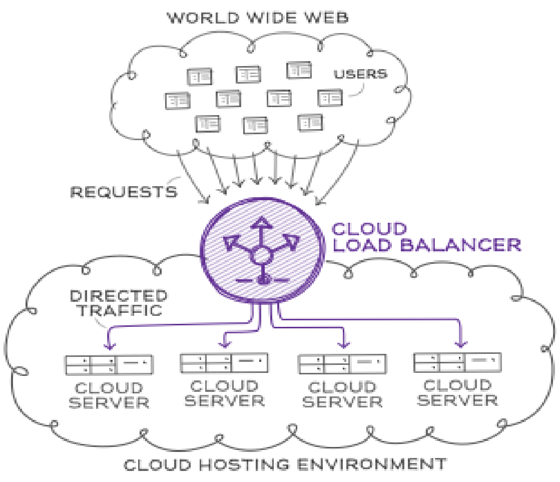

Figure 3 Load Balancing in Cloud

4.1. Goals of Load Balancing

A load balancing algorithm should satisfy the following goals [19]:

- Performance should be improved.
- Should have a backup even if the system fails partially.
- Should maintain the stability of the system.
- Should be upward compatible.

In order to balance the requests of the resources it is important to recognize a few major objectives of load balancing algorithms [15]:

- **Cost effectiveness:** The primary aim is to achieve an overall improvement in system performance at a reasonable cost.

- **Scalability and flexibility:** The distributed system in which the algorithm is implemented may change in size or topology. So the algorithm must be scalable and flexible enough to allow such changes to be handled easily.
- **Priority:** Prioritization of the resources or jobs needs to be done on beforehand through the algorithm itself. This helps in providing a better service to the important or high prioritized jobs in spite of equal service provision for all the jobs regardless of their origin.

4.2. Types of Load Balancing Algorithms

Based on process origination the Load balancing algorithms can be classified as [14] [16]:

- **Sender Initiated:** In this type of technique the client sends request until a receiver is assigned to him. Here, if a processor has a high load, it will try to share it with another processor.
- **Receiver Initiated:** In this case the receiver sends a request to acknowledge a sender who is ready to share the workload. If a processor has a low load it offers its services to the distributed system.
- **Symmetric:** In such type of algorithms it is a combination of both sender and receiver initiated type of load balancing techniques.

Based on the current state of the system the algorithms can be classified as [16]:

- **Static Load Balancing:** Static load balancing algorithms decide how to distribute the workload according to a prior knowledge of the problem and the system characteristics. Static load balancing algorithms are not pre-emptive and therefore each machine has at least one task assigned for itself. Its aims in minimizing the execution time of the task and limit communication overhead and delays. This algorithm has a drawback that the task is assigned to the processors or machines only after it is created and that task cannot be shifted during its execution to any other machine for balancing the load.
- **Dynamic Load Balancing:** Dynamic algorithms use state information to make decisions during program execution. An important advantage of this approach is that its decision for balancing the load is based on the current state of the system which helps in improving the overall performance of the system by shifting the load dynamically.

V. PARAMETERS FOR LOAD BALANCING

The different parameters based on which the load balancing algorithms should be designed are [12]:

- **Throughput:** A high throughput is required for better performance of the system.
- **Associated Overhead:** A successful implementation of the algorithm is the one which has minimum overhead.
- **Fault Tolerant:** The algorithm should perform correctly even though there is a failure of an arbitrary node.
- **Migration Time:** The time taken to move the tasks from one machine to another should be minimum by which the performance of the system can be improved.
- **Response Time:** It is the minimum time that a distributed system executing a specific load balancing algorithm takes to respond.
- **Scalability:** It is the ability of the system to accomplish load balancing algorithm with the limited number of resources.
- **Resource Utilization:** A good load balance algorithm should provide maximum resource utilization.
- **Performance:** The load balancing algorithm should be designed in such a way that it should improve the system performance.

VI. DYNAMIC LOAD BALANCING

6.1. Honeybee Foraging

This algorithm derived from the behaviour of honey bees. The honey bees use a procedure for finding and reaping food that is also applicable on distributed system to balance the load. There is a class of bees known as forager bees. These forage for food sources and after finding one they come back to give the idea of the quality or quantity of food with its distance from the beehive. The class of bees known as Scout bees then follows the foragers to reap it. After return, they give an idea of how much food is remaining. This results in more exploitation or abandonment of the food source. It is a decentralized load balancing technique based on natural procedure. System performance is enhanced with increased system dissimilarities but throughput is not increased with an increase in system size.

6.2. Biased Random Sampling

This algorithm is based on the construction of the virtual graph having connectivity between the all nodes of the system where each node of the graph is corresponding to the node computer of the cloud system. Edges between nodes are of two types. One is incoming edge and the other is outgoing edge. These are used to consider the load of particular system and allot the resources to the node. It is scalable technique to balance the load of the cloud system. It is also reliable and effective load balancing approach that is mainly developed to balance the load of distributed system.

6.3. Active Clustering

It is a self-aggregation load balancing technique which is used to optimize job assignments by connecting similar services using local re-wiring. The performance of the system is enhanced with high resources thereby increasing the throughput by using these resources effectively. It is degraded with an increase in system diversity.

6.4. Join-Idle Queue

This load balancing algorithm is used for dynamically scalable web services. This algorithm provides large-scale load balancing with distributed dispatchers. It works in two steps. First, it load balances the idle processors across dispatchers, for the availability of idle processors at each dispatcher. Then, it assigns jobs to the processors to reduce average queue length at each processor. By removing the load balancing work from the critical path of request processing, it effectively reduces the system load, incurs no communication overhead at job arrivals and does not increase actual response time.

6.5. Dynamically Scheduled Load Balancing using Boundary Value

The job is scheduled to the processing servers and the processed jobs will be sent to the client interface for the knowledge of the data processed. This job is mainly used in heterogeneous cloud computing system.

VII. STATIC LOAD BALANCING

7.1. Round Robin Algorithm

This algorithm distributes jobs evenly to all slave processors in a Round Robin fashion. The processor selection is performed in series and will be back to the first processor if the last processor has been reached. Processors selections are performed locally on each processor, independent of allocations of other processors. The main advantage of Round Robin algorithm is that it does not require inter process communication. In general Round Robin is not expected to achieve good performance in general case. However when the jobs are of unequal processing time this algorithm suffers as the some nodes can become severely loaded while others remain idle. Round Robin is generally used in web servers where generally HTTP requests are of similar nature and thereby be distributed equally. [2]

7.2. Central Manager Algorithm

Central Manager Algorithm, in each step, central processor will choose a processor to be assigned a job. The chosen slave processor is the processor having the least load. The central processor is able together all slave processors load information; thereby selection of the processor is made on this information. The load manager makes load balancing decisions based on the system load information. High degree of inter-process communication could make the state a bottleneck state. This algorithm is expected to perform better than the parallel applications, especially when dynamic activities are created by different hosts. [2]

7.3. Randomized Algorithm

Randomized algorithm uses random numbers to choose slave processors. The slave processors are chosen randomly following random numbers generated based on a statistic distribution. Randomized algorithm can attain the best performance among all load balancing algorithms, only for special purpose applications. [3]

7.4. Threshold Algorithm

The processes are assigned immediately upon creation to hosts. Hosts for new processes are selected locally without sending remote messages. Each processor keeps a private copy of the system's load. The load of a processor can characterize by one of the three levels: under loaded, medium and overloaded. Thresholds algorithms have low inter process communication and a large number of local process allocations. The later decreases the overhead of remote process allocations and the overhead of remote memory accesses, which leads to improvement in performance. A disadvantage of the algorithm is that all processes are allocated locally when all remote processors are overloaded. A load on one overloaded processor can be much higher than another overloaded processor, causing significant disturbance in load balancing, and increasing the execution time of an application.
Table 1 gives a brief comparison of the different load balancing algorithms.

Table 1 Comparative Analysis of Load Balancing Algorithms

Parameters	Round Robin	Central Manager	Randomized	Threshold	HoneyBee Foraging	Biased Random	Active Clustering	Join Idle Queue
Nature	Static	Static	Static	Static	Dynamic	Dynamic	Dynamic	Dynamic
Overload Rejection	No	No	No	No	Yes	Yes	Yes	Yes
Reliability	Less	Less	Less	Less	More	Moe	More	More
Stability	More	More	More	More	Less	Less	Less	Less
Fault Tolerant	No	Less	No	No	More	More	More	More
Resource Utilization	Less	Less	Less	Less	More	More	More	More
Response Time	More	More	More	More	More	More	More	Less
Waiting Time	Less	Less	Less	Less	More	More	More	More
Turnaround Time	Less	Less	Less	Less	More	More	More	More
Execution System	Decentralized	Centralized	Decentralized	Decentralized	Decentralized	Decentralized	Decentralized	Decentralized
Throughput	Low	Low	Low	Low	Low	Low	Low	High
Cost	Less	Less	Less	Less	More	More	More	More

VIII. Conclusion and Future Work

Load balancing is one of the main challenges in cloud computing. It is required to distribute the workload evenly across all the nodes. This helps to achieve a high user satisfaction and resource utilization ratio by making sure that every computing resource is distributed efficiently and fairly. Load balancing algorithms are totally dependent upon in which situations workload is assigned, during compile time or execution time. The above comparison shows that static load balancing algorithms are more stable than dynamic. Dynamic load balancing algorithms are always better than static in terms of overload rejection, reliability, adaptability, cooperativeness, fault tolerant, resource utilization, response time, waiting time and throughput. All the above algorithms have more advantages in terms of client execution time and cost.
In future, we need to develop a load balancing technique which is an advantage at the cloud service provider. This algorithm should be able to benefit the provider in terms of the cost. This can be done

by developing a load balancing algorithm that reduced the service cost at the cloud service provider by a certain standard amount irrespective of the services.

REFERENCES

[1]. Luis M. Vaquero, Luis Rodero Merino, Rajkumar Buyya, “Dynamically Scaling Applications in the Cloud”, ACM SIGCOMM Computer Communication Review, January 2011.

[2]. Abhijit A. Rajguru, S.S.Apte, “A Comparative Performance Analysis of Load Balancing Algorithms in Distributed System using Qualitative Parameters”, International Journal of Recent Technology and Engineering (IJRTE) ISSN: 2277-3878, Volume-1, Issue-3, August 2012.

[3]. André B. Bondi, “Characteristics of Scalability and Their Impact on PerformanceAT&T Labs, Network Design and Performance Analysis Department, ACM 2000.

[4]. Ossama Othman, and Douglas C. Schmidt, “Optimizing Distributed System Performance via Adaptive Middleware Load Balancing”, Department of Electrical and Computer Engineering, University of California, Irvine, Darpa Contract 2002.

[5]. Manjula K A, Karthikeyan P.,“Distributed Computing Approaches for Scalability and High Performance”, International Journal of Engineering Science and Technology Vol. 2(6), 2010.

[6]. Baldev Singh, Dr. O.P. Gupta, Simar Preet Singh, “Performance Evaluation of DNS BasedLoadBalancing Techniques for Web Servers”, IJCST Vol. 2, Issue 1, March 2011.

[7]. T.R.V. Anandharajan and Dr. M.A. Bhagyaveni, “Cooperative Scheduled Energy Aware Load-balancing technique for an Efficient Computational Cloud”, IJCSI International Journal of Computer Science Issues, Vol. 8, Issue 2, March 2011.

[8]. Sandeep Kumar Goyal and R.B. Patel, “Adaptive and Dynamic Load Balancing Methodologies For Distributed Environment: A Review”, International Journal of Engineering Science and Technology (IJEST), March 2011.

[9]. Alcides Calsavara and Luiz A. P. Lima Jr, “Scalability of Distributed Dynamic Load Balancing Mechanisms”, Graduate Program on Computer Science (PPGIa) Pontifical Catholic University of Paraná – PUCPR Curitiba, Brazil, The Tenth International Conference on Networks. ICN 2011.

[10]. A.Osman and H Ammar, ”Dynamic Load Balancing Strategies for Parallel Computers”, Department of computer science and electrical engineering West Virginia University, PO Box 6104, Morgantown. International Symposium on Parallel and Distributed Computing, 2002.

[11]. Amit Chhabra, Gurvinder Singh, Sandeep Singh Waraich, Bhavneet Sidhu, and Gaurav Kumar , “Qualitative Parametric Comparison of Load Balancing Algorithms in Parallel and Distributed Computing Environment”, World Academy of Science, Engineering and Technology, 2006.

[12]. Nayandeep Sran and Navdeep Kaur, “Comparative Analysis of Existing Load Balancing Techniques in Cloud Computing", International Journal of Engineering Science Invention ISSN (Online): 2319 – 6734, ISSN (Print): 2319 – 6726 www.ijesi.org Volume 2 Issue 1 ǁ January. 2013 ǁ PP.60-63.

[13]. Jeffrey M. Galloway, Karl L. Smith, Susan S. Vrbsky, “Power Aware Load Balancing for Cloud Computing”, Proceedings of the World Congress on Engineering and Computer Science 2011 Vol. I WCECS 2011, October 19-21, 2011, San Francisco, USA.

[14]. Aarti Ketan, Vivek Bhushan, Subhash Chand Gupta, “A Novel Survey on Load Balancing in Cloud Computing”, International Journal of Engineering Research and Technology, ISSN: 2278-0181, Vol-2, Issue-2, February-2013.

[15]. Sowmya Ray and Ajanta De Sarkar, "Execution Analysis of Load Balancing Algorithms in Cloud Computing Environment", International Journal on Cloud Computing: Services and Architecture (IJCCSA), Vol.2, No.5, October 2012.

[16]. Ms. Parin. V. Patel, Mr. Hitesh. D. Patel, Asst. Prof. Pinal. J. Patel, "A Survey On Load Balancing In Cloud Computing", International Journal of Engineering Research & Technology (IJERT) Vol. 1 Issue 9, November- 2012 ISSN: 2278-0181.

[17]. Y.Ranjit Kumar, M. Madhu Priya, K. Sahu Chatrapati, “Effective Distributed Dynamic Load Balancing for the Clouds”, International Journal of Engineering Research & Technology (IJERT), ISSN: 2278-0181, Vol. 2 Issue 2, February- 2013.

[18]. Walker, B. and D. Steel, 1999, “Implementing a full single system image unixware cluster: Middleware vs underware”, International Conference on Parallel and Distributed Processing Techniques and Applications PDPTA’99), June 1999, Monte Carlo Resort, Las Vegas, Nevada, USA. pp: 1-7.

[19]. 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.

AUTHORS

Nagamani H. Shahapure received the BE degree in Computer Science & Engineering from Gulbarga University and M.Tech degree in 2009 in Computer Network & Engineering from VTU, Belgaum. She is currently pursuing PhD in Cloud Computing in VTU under the Guidance of Dr.P. Jayarekha. She has 3 years of industry experience and 12 years of teaching experience. Currently she is working as an Assistant Professor in JSSATE

.

P Jayarekha holds M.Tech (VTU Belgaum) in Computer Science securing second rank and PhD in Computer Science. She has two decades of experience in the teaching field. She has published many papers. Currently she is working as an Associate Professor in the department of Information Science and Engineering at BMS College of Engineering, Bangalore.