Global Journal of Computer Science and Technology: B
Cloud and Distributed
Volume 18 Issue 1 Version 1.0 Year 2018
Type: Double Blind Peer Reviewed International Research Journal
Publisher: Global Journals
Online ISSN: 0975-4172 | Print ISSN: 0975-4350 | DOI: 10.17406

Load Balancing in Cloud Computing: A Survey on Popular Techniques and Comparative Analysis

By Rajgopal K T, Dr. K R. Anil Kumar & Nagesh Shenoy H
Visvesvaraya Technological University

Abstract- Cloud Computing is universally accepted as the most intensifying field in web technologies today. With the increasing popularity of the cloud, popular website's servers are getting overloaded with high request load by users. One of the main challenges in cloud computing is Load Balancing on servers. Load balancing is the procedure of sharing the load between multiple processors in a distributed environment to minimize the turnaround time taken by the servers to cater service requests and make better utilization of the available resources. It greatly helps in scenarios where there is misbalance of workload on the servers as some machines may get heavily loaded while others remain under-loaded or idle. Load balancing methods make sure that every VM or server in the network holds workload equilibrium and load as per their capacity at any instance of time. Static and Dynamic load balancing are main techniques for balancing load on servers. This paper presents a brief discussion on different load balancing schemes and comparison between prime techniques.

Keywords: load balancing, dynamic resource allocation, cloud load balancing techniques.

GJCST-B Classification: H.3.m

Strictly as per the compliance and regulations of:

Load Balancing in Cloud Computing: A Survey on Popular Techniques and Comparative Analysis

Rajgopal K T [α], Dr. K R. Anil Kumar [σ] & Nagesh Shenoy H [ρ]

Abstract- Cloud Computing is universally accepted as the most intensifying field in web technologies today. With the increasing popularity of the cloud, popular website's servers are getting overloaded with high request load by users. One of the main challenges in cloud computing is Load Balancing on servers. Load balancing is the procedure of sharing the load between multiple processors in a distributed environment to minimize the turnaround time taken by the servers to cater service requests and make better utilization of the available resources. It greatly helps in scenarios where there is misbalance of workload on the servers as some machines may get heavily loaded while others remain under-loaded or idle. Load balancing methods make sure that every VM or server in the network holds workload equilibrium and load as per their capacity at any instance of time. Static and Dynamic load balancing are main techniques for balancing load on servers. This paper presents a brief discussion on different load balancing schemes and comparison between prime techniques.

Keywords: load balancing, dynamic resource allocation, cloud load balancing techniques.

I. Introduction

a) Fundamentals of Load Balancing

A workload is of many types in real time and can be segregated into various categories like CPU load, network load, memory capacity issue, etc. While talking about cloud systems, load balancing mechanisms are used to share the load among virtual machines accessed by user nodes (end user devices) to improve resource utilization of the servers, enhancing the quality of service and providing high satisfaction to the users. Due to load sharing, every available processor in the cloud can work efficiently and smooth operations can be performed reducing delays [1]. Load balancing refers to the distribution of incoming load or tasks equally among the cloud nodes to achieve a good QoS (Quality of Service) by reducing response time and maximize resource utilization [2]. The dynamic load balancing algorithm uses system information while distributing the load [3]. A dynamic scheme is more flexible and fault tolerant. Load balancing enables advance network facilities and resources to offer better response and performance. Several algorithms are used to balance service requests or cloud data among nodes.

Cloud providers handle entire user requests load for smooth provisioning of services. Therefore, cloud service provider (CSP) uses numerous techniques for balancing the load. Load balancing is usually applied on a large amount of data traffic and servers to distribute work. Advanced architectures in the cloud are adopted to achieve speed and efficiency. There are several characteristics of load balancing such as equal division of work across available processors, facilitation in achieving the satisfaction of clients, improvement in end-to-end efficiency of the architecture, faster response time, and appropriate service allocations to achieve complete resource utilization [4]. There are two types of load balancing on the whole, i.e., Static and dynamic load balancing. Dynamic load balancing is used to rebalance the system being in running state when any overloaded VMs are detected. While static load balancing is used to balance the system at the starting phase by scheduling jobs to VMs. Static load balancing is generally chosen for the work as it avoids VM migration costs and delivers better quality of service (QoS) and lowers execution time [5]. Load balancing is considered as one of the most critical aspects to enhance the overall operational efficiency and performance of the cloud computing-based service provider. Balancing the oncoming load of the virtual machines equally among available resources implies that any of the running VMs doesn't stay idle or even partially loaded while other machines are facing heavy load. On the contrary, one of the critical challenges of cloud computing is to share and distribute the given workload dynamically among the available resources. The advantages of allocating the workload to available machines incorporate the expanded use of available resources which helps in improving the general performance, through which, greater customer fulfillment can be achieved. In this paper, we provide a comprehensive overview of interactive load balancing algorithms in cloud computing. Each algorithm addresses different problems from different aspects and offers diverse solutions. Some limitations of existing algorithms are performance issue, elevated processing

Author α ρ: Canara Engineering College/Computer Science & Engineering, Mangalore, 574219, Visvesvaraya Technological University, India. e-mail: rajgopalkt@gmail.com

Author σ: Dayananda Sagar College of Management & Information Technology /Department of MCA, Bangalore, 560078, Bangalore University, India.

time, starvation and limited to the environment where load variations are few, etc. The characteristic of an efficient load balancing algorithm dictates that it should be able to avoid overloading of any particular node. The main objective of load balancing is to assess the overall performance of the cloud computing systems in conjunction with the load balancing algorithms [6].

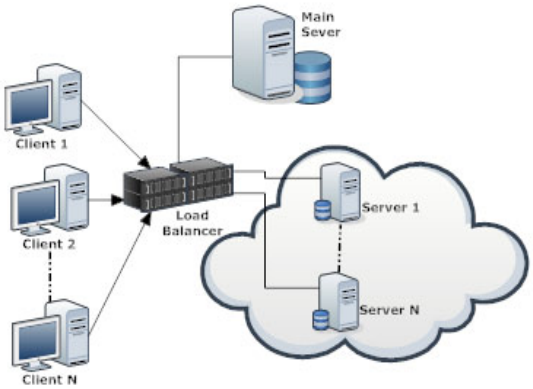

Figure 1: Load Balancing

b) Objectives of Load Balancing Schemes [7-8]

- To provide a noticeable improvement in the performance.
- To include a backup plan to overcome partial or complete system failure.
- To sustain a stabile system environment.
- To be able to adapt for future modifications and scalability in the system.

c) Benefits of load balancing [9]

- *Increased Scalability:* If you have a website, you must be uploading engaging content to attract readers. And it must be exciting to see a growing number of visitors on your site. However, it is significant to recall that the total volume of traffic on the website has a direct impact on the general performance of the website. If there is a sudden spike in the traffic, it might become difficult for your server to handle the excess traffic and the website may crash. Load balancing can help in spreading the traffic across multiple servers, and the increase in the traffic can be handled in a much easier manner. Depending on how the site's traffic fluctuates, the server administrators can scale the web servers up or down depending on your site's needs.
- *Redundancy:* The probability of hardware failure can be reduced, and the overall uptime of the website can be improved if load balancing mechanisms are used for maintaining a website or web application on more than one web server. By implementing load balancing you can achieve redundancy. This means that when the oncoming traffic to a website is distributed to more than two web servers, and if one of the servers goes offline, then the load balancer will spontaneously divert the traffic to the other online servers. When you maintain multiple load balanced servers, you can be assured that at least one server will constantly be online to control and respond to the site traffic even when there is hardware fail.
- *Reduced Downtime, Increased Performance:* You can schedule the maintenance and planned downtime at non-working hours like early mornings or the weekends, if your company is located in just one place. However, if you have a global business with offices scattered across the world, with different time-zones, you need to implement load balancing. This operation will enable you to shut off any server for maintenance and channel traffic to your other resources without disrupting work in any location. This way you can reduce the downtime, maintain the uptime and improve the performance.
- *Efficiently Manages Failures:* Load balancing helps in detecting failures early on and manages them efficiently, making sure that failure of any kind doesn't affect the servers or the workload. One can bypass the detected failures by re-distributing the available resources to other areas which are unaffected with the use of several data centers that are spread across a number of cloud providers. This mechanism will reduce the disruption and failures.
- *Increased Flexibility:* IT administrators can enjoy great flexibility in handling website traffic by using multiple load balanced servers. They can perform several maintenance tasks on the server without affecting the total uptime of the website. This operation is accomplished by diverting the entire traffic to any one server and keeping the load balancer in active or passive mode. You have the flexibility of having a staggered maintenance system, where at least one server is always available to pick up the workload while others are undergoing maintenance. This method ensures that the users of the website do not suffer from any outages at any time.

II. Load Balancing: Taxonomy

a) Static Load Balancing

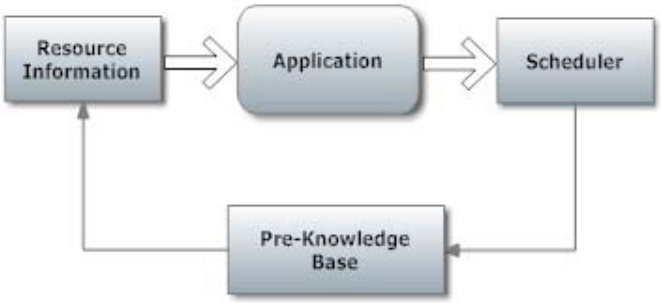

Figure 2: Static Load Balancing

Most of the static load balancing techniques are non-preemptive in nature. It implies that the load once assigned to a node, it cannot be diverted to any other node [10] [11]. The execution time required for this method is low as it has lesser communication. One of the biggest shortcoming of this technique poses a major threat to the overall performances of the system. This happens because of the load fluctuation in the distributed system. The static load balancing algorithms can be classified into four types: central manager, round-robin algorithm, randomized algorithm and threshold algorithm [12].

- *Central Manager Algorithm [14]:* In this technique, a central node will be designated to choose a slave node for assigning the load. The slave with lowest load will be chosen and the job will be assigned. The central node holds the load index for all slave nodes which are associated with it. In a situation where the load is varied, the slave nodes will transmit a message to the central node. On the downside, this algorithm leads to bottleneck because it requires larger inter-process communication. The performance of this algorithm is better for dynamic activities of various hosts.
- *Round Robin Algorithm [13]:* This algorithm distributes the load equally among all the nodes. The work-load distribution method is termed as round robin, wherein, equal load will be assigned to the node which moves about in a circular fashion without any priority. It forms a circular structure and hence the load will come back to the first node and this process continues. The nodes will hold its load index which are unrelated to allocations from the remote node. The benefits of round robin are that, it is starvation free, simple and easy to implement. Since it doesn't need any inter-process communication, this algorithm provides better performance for special purpose applications. On the downside, this algorithm doesn't work perfectly when the given general-purpose jobs have unequal execution time.
- *Randomized Algorithm [16]:* In this strategy, the selection of node is made randomly without any knowledge on the current or previous load taken by the node. It is suitable for conditions where the system has equal load on each node because it is static in nature. The performance of this algorithm is found to be better for special purpose applications. The inter-process communication is not needed as the nodes maintains their own load record. In certain circumstance, a single node can be overloaded, while other nodes are still underloaded.
- *Threshold Algorithm [15]:* In this technique, when a new node is created, the load will be instantly assigned to it. The selection of nodes are made locally without any transmission of remote messages. A private copy of load will be maintained by each node. The characterization of load is segregated into under-load, medium-load and over-load.

b) Dynamic Load Balancing

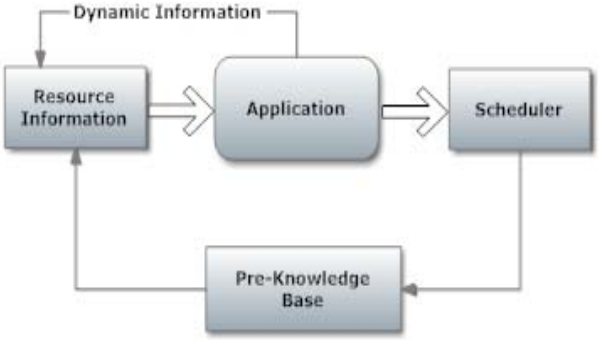

Figure 3: Dynamic load Balancing

The Dynamic loads balancing technique will supervise the alterations on the workload of the system, and it will automatically reorganize the workload appropriately. The process is carried out in three methods, i.e., transfer strategy, location strategy, and information strategy. Qualifying tasks for load transfer will be decided by the Transfer strategy, which selects the tasks that can be processed by other nodes. The remote node which should process the transferred task will be selected using Location strategy. Information strategy dictates the transfer strategies and location of nodes, and hence, it acts as an information center for load balancing algorithm.

The dynamic algorithms are designated to operate in three different controlling forms, which are termed as, distributed, semi-distributed and centralized. In distributed form, the responsibility will be segregated equally among all nodes. In a semi-distributed scenario, the network will be sub-divided into smaller clusters, in which each cluster will be centralized. In centralized load distribution scheme, a single node will be nominated as

a central node of the network which will be responsible for carrying out the task of load distribution. With the cooperation of central nodes present in the clusters, the load balancing of system can be established.

- *Centralized Strategy [18]:* In a centralized Strategy, the load balancer will be positioned on one principal workstation node. Some of the basic characteristics of centralized strategy is:
- The list of tasks to be performed will be maintained by a master node.
- The tasks will be forwarded to the executing node
- Once the process completes a task, a request is made to the master node for another task.

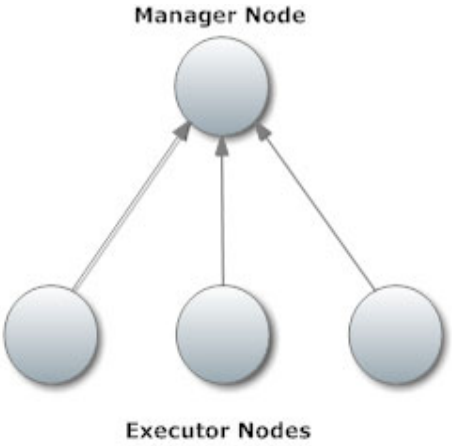

Figure 4: Centralized Strategy

- *Decentralize Strategy[19]:* In a decentralized technique, the load balancer is reproduced on all workstations, which then allocates the tasks to the nodes. For this purpose of job selection, the decentralized technique uses various algorithms, such as random polling algorithm, round robin algorithm, etc. Whenever a node fails, the operational nodes can take-over the task.

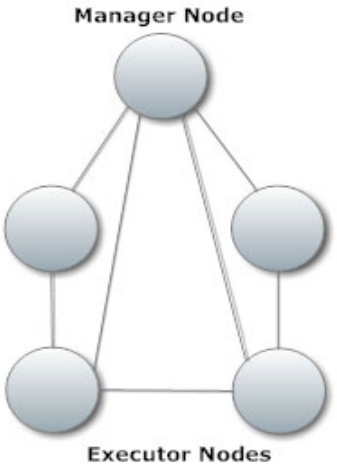

Figure 5: Decentralized Strategy

- *Ant Colony Algorithm [20]:* The ant colony algorithm maintains the documentation and record of every node that it visits. This information is helpful for efficient decision-making efforts in the future. To select a next node, an ant will deposit pheromones during their movement. Various factors can affect the intensity of pheromones, such as distance of food, type of food, quality of food etc. The pheromones are updated as soon as the job completes successfully. An individual result set will be generated by each ant, which are then combined to construct a complete solution. Instead of updating their own result set, the ant will constantly update a single result set. Later, the solution set is updated by the ant pheromones trials.
- *Throttled Algorithm [21][22]:* Throttle Algorithm takes its inspiration from virtual machines. In this approach, the client or user makes a request to the load balancer to determine a virtual machine which is capable of handling the load easily and processes the operations and task laid out by the user. A table of virtual machines and their states (available or busy), will be maintained by the load balancer. Hence, the client will request to the load balancer to select an appropriate Virtual Machine to execute the obliged operations.
- *Honey Bee [23]:* This algorithm has got inspiration from the behavior of honey bees. This load balancing algorithm is dynamic in nature. When it comes to honey bees, they can be categorized into two types, the finders and the reapers. The finders are responsible for searching the source of honey. When a source for honey is found, the finders perform a waggle dance which can tell about the quantity and quality of the honey. Later, the reapers will travel and collect the honey from these sources. They in turn perform waggle dance to signal the amount of honey left in the source.
- *Active Clustering Load Balancing Algorithm [24]:* This algorithm is works by segregating and grouping similar nodes together. The further operation is based on these grouped nodes. The advantage of this grouping of nodes is that, it helps the resources to boost the throughput cost-effectively. A term match-maker is coined in this algorithm. It all begins when a node chooses the neighboring node, which in turn selects and matches the initial node with its neighboring node, which has similar characteristics. After successful connection, the match-maker node will be discarded. This process is repeated until the load is balanced equally. The throughput is considerably increased with this algorithm which improves the system performance and promises an efficient utilization of resources.

- *CARTON [25]:* CARTON is an amalgamation of load balancing (LB) and distributed rate limiting (DRL). Through load balancing, jobs are fairly assigned to the servers. DRL ensures the equal distribution of resources. Thus, the workload is dynamically assigned to improve the performance and spread the load equally to all the servers. This algorithm can easily be implemented as low communication required.

III. Literature Survey

In 2012, Rattan Mishra [26] introduced "Ant colony optimization" (ACO) to avoid deadlock condition in cloud computing systems. The implementation carried out on two different job scheduling strategies, i.e., time shared and space shared. According to acquired experimental results it consumed less memory during processing of tasks as previously implemented resources and provided high performance.

In 2014, Stuti Dave et al. [27] presented a "Round Robin" (RR) for load balancing at the virtualized environment. In this paper, they have suggested improved Fair RR algorithm approach that provides dynamic time quantum strategy. When the request enters the ready queue, they are processed and calculated according to time quantum and burst time computation while VM's are allocated. Thus, FRR provide fairness to larger and smaller incoming requests at executing load resulting in faster load balancing in the cloud.

In 2012, O.M Elzeki et al. [28] has proposed upgraded "Max-Min" algorithm to escalate the Max-Min efficiency by synchronized parallel execution of task as resources and emphasizes on selecting task with maximum completion time. This algorithm computes and estimates the projected completion time of the presented tasks on each resource. With the help of this information, the projected execution time will be allocated to a resource that has the least completion time. As a final point, the scheduled task will be detached from the Meta-tasks and all the estimated times will be updated. This process will be reiterated until all the submitted tasks are executed.

In 2011, T. kokilavani [29] proposed "Load balance Min-min" (LBMM) algorithm, which is basically a grid scheduling algorithm. The Min-Min will be executed in the first round. In the next round, it picks out the resources which are bearing heavy load and reallocates them to the resources which have lesser load. It later detects the resources with high make span and then selects the tasks with lower execution time present on that resource. A comparison will be made between the make span produced by Min-Min and the completion time. If it is found to be lower, then the task is rescheduled, otherwise next higher completion time of task will be chosen. This process continues until all the resources and tasks are completely exploited.

Singh et al. [30] propositioned a novel load balancing algorithm which is known as the Vector Dot. The problems relating to hierarchical complexity of the datacenters were addressed with the help of this algorithm. It also handles the multi-dimensionality of resource loads across various servers and network switches. It extends the support to the storage in an agile data center which contains an integrated server with virtualized storage system.

Stanojevic et al. [31] proposed a technique called CARTON that amalgamates the usage of LB and DRL for cloud control. The LB (Load Balancing) is implemented to distribute the jobs equally among various available servers with an intention to minimize the associated costs and DRL (Distributed Rate Limiting) is deployed to ensure that the resources are distributed in a particular manner so that a fair resource allocation and utilization is established.

Y. Zhao et al. [32] have focused and tackled the problem of intra-cloud load balancing among physical hosts by using the technique of adaptive live migration of virtual machines. The idea of load balancing model is conceived and implemented to lower the migration time of virtual machines in a shared storage environment, in order to balance they applied load on the servers according to their capacity and processor, memory or IO utilization.

V. Nae et al. [33] proposed an Event Driven Load Balancing Algorithm (EDLBA) for a real-time Massively Multiplayer Online Games (MMOG). The algorithm operates by taking input in the form of capacity event and analyzes the components in the perspective of the resources and global state of the game session. Then later, it generates the output actions in the form of the game session load balancing act.

J. Hu et al. [34] presented a scheduling mechanism on load balancing of Virtual machines resources which uses the historical data and current state of the system. With the help of a genetic algorithm, the proposed technique provides the best load balancing and minimized dynamic migration.

A. Bhadani et al. [35] proposed a strategy to evenly balance the load in a distributed virtual machine or cloud computing environment, known as Central Load Balancing Policy for Virtual Machines (CLBVM).

H. Liu et al. [36] proposed a technique which can provide a large scale net data storage model in collaboration with the Storage as a Service model based on Cloud Storage system. This technique is termed as load balancing virtual storage strategy (LBVS). A three-layered architecture and load balancing technique with two load balancing modules is employed to achieve Storage virtualization, which is known to enhance the overall efficiency.

The Y. Fang et al. [37] examined load balancing and conferred a two-level task scheduling mechanism to meet the dynamic requirements of users and achieve better resource utilization. The working of the algorithm is quite simple. It maps the tasks on to the virtual machines, which are then assigned to the hosts depending on the available hardware resources. With this approach, the resource utilization is better and the response time for a task is improved, which in turn enhances the overall performance of cloud computing.

M. Randles et al. [38] inspected a nature-inspired algorithm, in particular, the decentralized honey-bee based load balancing technique for the purpose of providing self-organization. In the course of local server actions, the algorithm is able to accomplish global load balancing. The performance of the cloud system is enriched with higher system diversity. On the contrary, the increment in system size will not maximize the throughput. Hence, this type of algorithm is suitable for certain situations in which requires diverse population of service types.

Y. Lua et al. [39] presented an algorithm for dynamically scalable web services, known as Join-Idle-Queue load balancing. This technique presents a large-scale load balancing schemes with distributed dispatchers. The process begins with identifying the idle processors across dispatchers and check for their availability. With this information, assigning jobs to processors helps in minimizing the average length of queue at each processor. When the load balancing task is removed from the critical path of request, the incurred load will be significantly reduced. Furthermore, the communication overhead at request arrival is also reduced and the response time will not increase.

Baris Yuce et al. [40] introduced an algorithm known as "Honey bee inspired algorithm" which centers on refining the benchmark functions, which are later compared with other optimization techniques such as PSO, ACO and EV. This provides an evaluation of bee behavior and algorithm. The primary objective of this technique is to enhance the bee's algorithm with the help of adaptive neighborhood sizes and site abandonment (ANSSA) mechanism.

Author [14] [39] [41], presented a novel algorithm on content aware load balancing policy, which was termed as workload and client aware policy (WCAP). This proposed technique employs a parameter known as USP to postulate a unique and distinctive property of the arriving requests and computing nodes. The scheduler makes use of this USP information to determine the finest appropriate nodes for processing the request.

IV. Comparison & Discussion

In this section we demonstrate various techniques used for load balancing and researchers proposed mechanisms. In Table 2 we compared various types of algorithms used for load balancing and illustrated their merits and demerits. In Table 3 of this section we illustrated the mechanisms proposed by various researchers and made a short comparison in terms of its efficiency.

Table 1: Comparison between Static and Dynamic Load Balancing

S.NO.	Factor	Load Balancing	
		SLB Algorithm	DLB Algorithm
1	Nature	Work load is assigned at compile time	Work load is assigned at run time
2	Overhead involved	Little overhead due to IPC	Greater overhead due to process redistribution
3	Resource utilization	Lesser utilization	Greater utilization
4	Processor thrashing	No thrashing	Considerable thrashing
5	State woggling	No woggling	Considerable woggling
6	Predictability	Easy to predict	Difficult to predict
7	Adaptability	Less adaptive	More adaptive
8	Reliability	Less	More
9	Response time	Short	Longer
10	Stability	More	Less
11	Complexity	Less	More
12	Cost	Less	More

Table 2: Load Balancing Algorithms Comparison

Algorithms	Static / Dynamic	Advantages	Disadvantages
Ant Colony	●	Information can be collected faster by the ants.	– Network is over headed so search takes longer time. – No clarity about the number of ants.
Round-Robin	★	– Chooses a random VM to allot a task. – This mechanism assigns the VMs, sorted in the queue with no consideration of job priority.	– The operation time of each process is not identified beforehand. – Forecast of operation time is unfeasible.
Min-Min	●	This method is most appropriate in situations where multiple jobs to be accomplished within least time.	It leads to the starvation problem.
Max-Min	●	This method is appropriate when the jobs are with highest completion time, as it eradicates the starvation.	The job, which should be finished in the least time needs to stay in prearranged queue till the jobs with highest completion time get over.
Central Manager	●	Works well when dynamic operations are initiated through diverse hosts.	Sometimes leads to System bottleneck state.
Threshold	★	Keeps the private copy of information in node.	Can't able to distribute information, if private copy node fail whole System will fail.
Randomized	●	Give the best performance.	At times, solitaire nodes may get burdened with tasks even as the other node is moderately laden.
Active Clustering	★	It balances the load pretty well.	Performs poorly in heterogeneous environment
Honey Bee	★	Increases throughput; Minimize response time.	High priority tasks can't work without VM machine.
Carton	★	– Fairness; Good performance; Equal distribution of responses. – Low communication is required.	It depends upon lower costs.
Throttle	★	Good performance as List is used to manage the tasks.	Tasks need to wait for longer time.
● - Static ★ - Dynamic			

Table 3: Comparison of Various Schemes

Ref.	Name	year	Mechanism	Advantages	Disadvantages
[27]	Stuti Dave et al.	2014	Round Robin for Cloud Virtualized Environment	Every process gets equal weight so no process will go under starvation	Most of the time processor Remains idle.
[28]	O.M Elzeki et al.	2012	Max-Min Approach	Requirements are prior known.	It takes longer time to complete the task.
[29]	T. kokilavani	2011	Load Balance Min-Min (LBMM)	Enhances the load misbalance of Min-Min approach and reduces the operation time.	It doesn't state node selection for a complex job that necessitates heavy calculations.
[30]	Singh et al	2008	Vector Dot	Good in Resource Utilization	Some time functions do not work properly.
[31]	Stanojevic et al	2009	Ant Colony Optimization	Faster information can be collected by the ants	No clarity about the number of ants.
[32]	Y. Zhao et al	2009	Intra-Cloud Load Balancing	It manages the node at the center	If inter-node fails, overall System fails.
[33]	V. Nae et al	2010	EDLBA	Receiving capacity events as input good performance.	Depends on time, if time exceeds then no reverse process takes place
[34]	J. Hu et al.	2010	SOLB	Mechanism performs well.	Waits for each task to process.
[35]	A. Bhadani et al	2010	CLBVM	Suitable for heterogeneous network	Priority is fixed and bottleneck problem
[36]	H. Liu et al	2010	LVBS	Storage virtualization achieved. Increase the efficiency	Virtualization fails not works properly.
[37]	Y. Fang et al	2010	Dual-Stage Job Scheduling	Improves resource Utilization.	Slow performance
[38]	M. Randles et al	2010	Decentralized Honey Bee	Performance of system increases	Sometimes System Bottleneck.
[39]	Y. Lua et al	2012	JIQLB	Reduces system's workload. Doesn't lead to communication overhead at job advent Works well under	More power consumption.
[40]	Baris Yuce	2013	Honey Bee Inspired Algorithm	heterogeneous resources Selects the most	Amplification of VMs doesn't enhance throughput evenly.
[14] [39] [41]	Lua Y	2011	Workload And Client Aware Policy (WCAP)	appropriate processor for catering the user requests.	If Client fails, then overall System fails

V. Conclusion

Load balancing is a procedure utilized to equally distribute the workload on available processors or VMs so that all machines share the workload and no processor is overloaded. Thus, it can be said that load balancing definitely increases throughput and reduces response time. Load balancing is done by static or dynamic load balancers that accept multiple requests from users and distributes them across servers on the cloud. Today efficient load balancing is one of the greatest concerns in cloud computing systems. To solve this issue, various techniques were proposed by researchers and experts. In this paper, we have classified types of load balancing and reviewed numerous literature about various existing load balancing methods in cloud computing environment by researchers in this field. We demonstrated the comparisons of popular mechanisms proposed by researchers in terms of efficiency and merits or demerits. Further, this survey can be extended to review various machine learning and genetic algorithm usage in load balancing arena.

References Références Referencias

1. Abraham, "Genetic algorithm based schedulers for grid computing systems Javier Carretero, Fatos Xhafa," in International Journal of Innovative Computing, Information and Control, vol. 3, no. 6, pp. 1–19,2007.
2. Sukhvir Kaur,Supriya Kinger,"Review on Load Balancing Techniques in Cloud Computing Environment", International Journal of Science and Research (IJSR),Paper ID: 02014812,Volume 3, Issue6, June2014, pg. 2499-2504
3. M. Katyal and A. Mishra, "A comparative study of load balancing algorithms in cloud computing environment." in International Journal of Distributed and Cloud Computing, Volume 1 Issue 2 December 2013.
4. R. G. Rajan and V. Jeyakrishnan, "A survey on load balancing in cloud computing environments," in International Journal of Advanced Research in Computer and Communication Engineering, vol. 2, no. 12, pp. 4726–4728, 2013.
5. Ramezani, J. Lu, and F. K. Hussain, "Task-Based System Load Balancing in Cloud Computing Using Particle Swarm Optimization," International Journal of Parallel Programming, vol. 42, no.5, pp. 739-754, 2013.
6. Ali M Alakeel, "A Guide To Dynamic Load Balancing In Distributed Computer Systems", International Journal of Computer Science and Network Security, Vol. 10 No. 6, June 2010.
7. David Escalnte and Andrew J. Korty, "Cloud Services: Policy and Assessment", EDUCAUSE Review , Vol. 46, no. 4, 2011.
8. Parin. V. Patel, Hitesh. D. Patel, Pinal. J. Patel, "A Survey on Load Balancing in Cloud Computing" IJERT, Vol. 1, Issue. 9, pp. 1-5, 2012.
9. https://www.edgehosting.com/managed-solutions/ managed hosting services/server load balancing/ benefits.
10. A. N. Tantawi and D. Tawsley, "Optimal Static Load Balancing in Distributed Computer Systems" Journal of the ACM, Vol. 32, No. 2, pp. 445-465, 1985.
11. S. H. Bokhari, "Dual Processor Scheduling With Dynamic Reassignment", IEEE Transactions on Software Engineering, Vol. SE-5, No. 4, pp. 341-349, 1979
12. Aarti Khetan, Vivek Bhushan, Subhash Chand Gupta, "A Novel Survey on Load Balancing in Cloud Computing,"International Journal of Engineering Research & Technology (IJERT), Vol. 2, Issue 2, 2013
13. Jasmin James, Dr. Bhupendra Verma, "Efficient VM Load Balancing Algorithm for a Cloud Computing Environment ", International Journal on Computer Science and Engineering (IJCSE) , Vol. 4 No. 09 Sep 2012 , pp. 1658-1663.
14. Gulshan Soni, Mala Kalra, "A Novel Approach for Load Balancing in Cloud Data Center", International Advance Computing Conference (IACC) IEEE, 2014, pp.807-81
15. Sharma Sandeep, Singh Sarabjit and Sharma Meenakshi (2008), "Performance Analysis of Load Balancing Algorithms", World Academy of Science, Engineering and Technology, 2008
16. Isam Azawi Mohialdeen, "Comparative Study of Scheduling Algorithms in Cloud Computing Environment ", Journal of Computer Science, 2013, pp. 252-263
17. Ruixia Tong and Xiongfeng Zhu, A Load Balancing Strategy Based on the Combination of Static and Dynamic, 2010 2nd International Workshop in Database Technology and Applications (DBTA), 2010, pp. 1-4.
18. Red hat: Red hat enterprise virtualization 3.2technical reference guide 22015. URL https://access.redhat.com/site/documentation/en-US/Red_Hat_Enterprise_Virtualization/3.2/ html/Technical_Refe-rence_Guide/index.html
19. Christodoulopoulos K, Sourlas V, Mpakolas I, Varvarigos E., A comparison of centralized and distributed meta-scheduling architectures for computation and communication tasks in grid networks. Computer Communications 2009; 32(7):1172–1184.
20. R.S. Chang, J.S.Chang, and P.-S. Lin, "An ant algorithm for balanced job scheduling in grids," Futur. Gener. Comput. Syst. , vol. 25, no. 1, pp. 20–27, Jan. 2009.
21. D. B. L.D. and P. Venkata Krishna, "Honey bee behavior inspired load balancing of tasks in cloud computing environments," Appl. SoftComput., vol. 13, no. 5, pp. 2292–2303, May 2013
22. M. Kawser "Performance comparison between round robin and proportional fair scheduling methods for LTE," Int. J. Inf. Electron.Eng., vol. 2, no. 5, 2012.
23. Hung-Chang Hsiao, et al., "Load Rebalancing for Distributed File Systems in Clouds", IEEE transactions on parallel and distributed systems, VOL. 24, NO. 5, MAY 2015.
24. Martin Randles, David Lamb, A. Taleb-Bendiab , "A Comparative Study into Distributed Load Balancing Algorithms for Cloud Computing ", IEEE 24th International Conference on Advanced Information Networking and Applications Workshops , 2010, pp. 551-55
25. J. Hu, J. 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 Architectures, Algorithms and Programming (PAAP), 2010.

26. Mishra R. and Jaiswal A., "Ant Colony Optimization: A Solution of Load Balancing in Cloud", International Journal of Web & Semantic Technology (IJWest), Vol 3, No 2, April 2012.
27. Stuti Dave, Prashant Mehta "Round Robin Concept for Load Balancing Algorithm at Virtual Machine Level in Cloud Computing" IJAC (0975-8887) Volume 94\-No.4, May 2014.
28. O.M. Elzeki, M.Z. Reshad, M.A. Elsoud "Improved Max- Min Algorithm in Cloud Computing" IJCA (0975-8887) Volume 50- No. 12 July 2012.
29. Kokilavani and Dr. D.I. George Amalarethinam, "Load Balanced Min-Min Algorithm for Static MetaTask Scheduling in Grid Computing", International Journal of Computer Applications (0975–8887) Volume 20–No.2, April 2011.
30. Singh A, Korupolu, M, Mohapatra D. Server-Storage Virtualization: Integration and Load Balancing in Data Centers. Proceedings of the 2008 ACM/IEEE conference on Super computing (SC'08); Austin, TX; 2008. p. 1–12.
31. R. Stanojevic, and R. Shorten, "Load balancing vs. distributed rate limiting: a unifying framework for cloud control", Proceedings of IEEE ICC, Dresden, Germany, August 2009, pages 1-6.
32. 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, August 2009, pages 170-175.
33. Nae V., Prodan R. and Fahringer T., 11th IEEE/ACM International Conference on Grid Computing (Grid), 9-17, 2010.
34. Hu J., Gu J., Sun G. and Zhao T., 3rd International Symposium on Parallel Architectures, Algorithms and Programming, 89-96, 2010.
35. Bhadani A. and Chaudhary S., 3rd Annual ACM Bangalore Conference, 2010.
36. Liu H., Liu S., Meng X., Yang C. and Zhang Y., International Conference on Service Sciences (ICSS), 257-262, 2010.
37. Fang Y., Wang F. and Ge J., Lecture Notes in Computer Science, 6318, 271-277, 2010
38. Randles M., Lamb D. and Taleb Bendiab A., 24th International Conference on Advanced Information Networking and Applications Workshops, 551556, 2010.
39. Y. Lua, Q. Xiea, G. Kliotb, 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 Performance Evaluation, vol. 68, pp.1056-1071, November 2011.
40. Yashpalsinh Jadeja and Kirit Modi, "Cloud Computing - Concepts, Architecture and Challenges", International Conference on Computing, Electronics and Electrical Technologies [ICCEET], IEEE-2012
41. Cristian Klein, Alessandro Vittorio Papadopoulos, Manfred Dellkrantz, Jonas Durango, Martina Maggio, KarlErik Arzen, Francisco Hernandez-Rodriguez, Erik Elmroth, Improving Cloud Service Resilience using Brownout-Aware Load-Balancing, Reliable Distributed Systems (SRDS), 2014 IEEE 33rd International Symposium, Oct 2014, Pages: 31-40.