Cloud Load Balancing Using Optimization Techniques

Ajay Jangra and Neeraj Mangla

Abstract Cloud computing is an integrated phenomenon that incorporates data, applications and services in a dynamic environment and enables worldwide optimization of resources. This computing technology is scalable and elastic in nature that opens door for large amount of incoming data from different venues with high velocity. Managing such data in distributive and heterogeneous environment imposes a challenge of load balancing on the service providers. They need to allocate the incoming tasks efficiently to the computing nodes to avoid imbalanced mapping and execution of the tasks. To achieve efficient load balancing, various load balancing algorithms have been proposed, and they all focus on achieving the effective distribution of data and improve the associated measurement factors. In this paper, different load balancing algorithms have been studied and analyzed with description of their techniques and focused parameters. Then, there is a brief discussion on the existing load balancing algorithms and further compares them based on parameters like throughput, scalability, resource utilization, etc., followed by the important findings thus made.

Keywords Ant colony · Agent-based algorithm · Cloud computing · Estimated finish time · Genetic algorithm · Load balancing · Honeybee foraging and throttled algorithm

A. Jangra (✉)
PhD scholar, Maharishi Markandeshwar (Deemed To Be University) MMDU, Mullana, Ambala, India
e-mail: er_jangra@yahoo.co.in

N. Mangla
CSE Department, Maharishi Markandeshwar (Deemed To Be University) MMDU, Mullana, Ambala, India
e-mail: erneerajynr@gmail.com

N. Marriwala et al. (eds.), *Mobile Radio Communications and 5G Networks*,
Lecture Notes in Networks and Systems 140,
https://doi.org/10.1007/978-981-15-7130-5_60

1 Introduction

Cloud computing is a modeled technique, which equipped users with virtualized pool of resources in distributive environment and also facilitated pay as you go model for the resource utilization [1]. Cloud service providers (CSP) aim to serve request online as per the type of cloud environment (public, private, hybrid and community) and these services. In the beginning of cloud era, IaaS, PaaS and SaaS were the three accessible elastic and scalable services of cloud computing. These are Internet-based rented services made available by CSPs and assured over subscription through service-level agreement (SLAs) [2]. The client/customer does not need to own the hardware or software rather than they can use them online with the Internet facility.

With the enhancement in computing era, the cloud computing is rapidly enriching and able to offer (EaaS) ***Everything as a Service***. Besides, day to day the amount of data evolving with it is also gaining an attention to be managed efficiently for the effective and sustainable working over distributed cloud scenario [3]. It is one of the major responsibilities of data handler to ensure the reliable and secure data handling in cloud. Large amount of data in cloud increase the velocity of demands which initiates a new challenge of load balancing in cloud computing that targets the effective scheduling and resource monitoring [4–6].

This paper justifies the requirement of load balancing in cloud computing with various supporting strategies that focus on improving the process of efficient resource utilization.

1.1 Load Balancing in Cloud Computing

Load balancing is a uniform approach of scheduling jobs among the available computing nodes. Monitoring resources and their effective utilization in broad network access is the main objective of load balancing [7]. A load balancing algorithm is considered resourceful when it is fault-tolerant and scalable in nature and guarantees to produce maximum throughput. There are numerous algorithms available that are categorized as dynamic or static in nature and operate in different environment. Static algorithms possess predefined factors and states on which it runs, whereas dynamic algorithms are operable at run time depending on the current state dynamically balancing the traffic on server. Various types of load balancing algorithms are as follows [6, 8].

A. **Static Load Balancing**

Static load balancing techniques are non-preemptive in nature that has predefined strict rules to be followed based on input and does not depend upon the current state of the machine to manage the workload. It requires a prior knowledge of the system setup and resource availability. It is also known as policy-driven load balancing driven by parameters like server capacity, throughput rate, fault tolerance, response time, etc. [9]. Examples of static load balancing technique are artificial bee colony search, two phase scheduling, central manager algorithm, etc.

B. **Dynamic Load Balancing**

Dynamic load balancing techniques are preemptive in nature that do not require prior knowledge of input and depend on the current state and enhance the overall working of the system. It manages load dynamically and prevents nodes from getting overloaded as it transfers load within nodes on run time. It is also known as feedback-driven load balancing [10]. Examples of dynamic load balancing technique are artificial ant colony search, round robin algorithm, throttled load balancing algorithm, etc.

1.2 Load Balancing Optimization Algorithms

This section shows the literature review done on different load balancing optimization algorithms describing different ways to balance huge data in cloud.

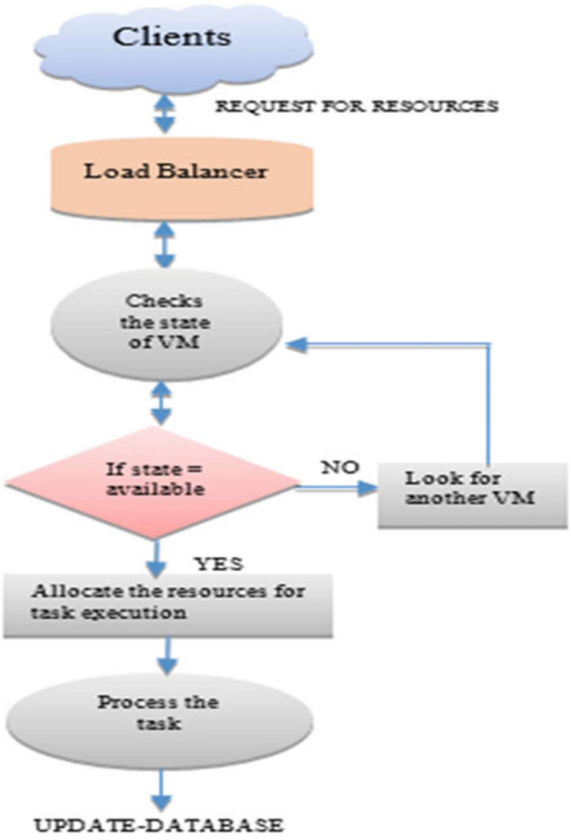
Throttled Load Balancing Algorithm

Throttled algorithm is a state-based algorithm that depends on the current state of virtual machines whether it is available or busy. Load balancers [11] are the modules of operating environment that dynamically balance the load on different available virtual machines and maintain their index as well as its associated state. If the state of virtual machine is available, the request is assigned; otherwise, the action is declined and searched for safe state. Virtual machines are responsible for the execution of request after its state has been verified by the load balancer. In [12], the comparison is made by the author between round robin and throttled algorithm in terms of time and cost. It is considered superior as compared to round robin algorithm in terms of cost as it reduces the cost of virtual machine's usage per hour (Fig. 1).

Ant Colony Optimization Algorithm

Ants are small blind insects that used to find food outside their nest when they are hungry. The way they find their food by following the shortest path is associated with load balancing in cloud. These ants decompose pheromone on their way to food with same speed and same rate. This helps the other ants to follow their path to food. So, more the followers, the higher will be the concentration of the pheromone decomposed. Evaporation rate of this pheromone on the shortest path is quiet low. In [13], the author has shown that this same process can be adopted by the job scheduler of cloud computing for mapping the task to available executing nodes. Schedulers check the load on their surrounding and transfer request accordingly for effective utilization by maintaining the descriptive table containing all the necessary information about virtual machines.

Fig. 1 Process flow of throttled algorithm



Honeybee Optimization Algorithm

It is one of the finest algorithms of load balancing in cloud computing that follows the behavior of honeybees. There are two categories of honeybees: One is detector honeybees that go out in the search of food and other one is follower honeybees that follow the path directed by leader honeybees. Leader honeybees come back and perform the well-known waggle dance in which they form numeric eight. This special dance tells the quality and quantity of the food and also the duration of the dance shows the distance of the food from beehive with its recorded profit [14]. Unemployed bees in the hive have the option to be detectors or followers. This paper [15] shows the improved artificial honeybee algorithm as the basic algorithm may create some imbalance of load among the nodes. According to this improved algorithm, threshold value is set for every server queue, and when the length of a server queue exceeds the value, the load is transferred to another server, and they are executed independently improving the throughput of the system.

Genetic Algorithm

Genetic algorithm is one of the optimal algorithms of effective search and optimization in load balancing. Simple GA follows the three-step process of selecting the population followed by genetic operations and replacement with new population. The genetic-based load balancing strategy in [16] balances the load based on process described based on genetic algorithm. It first initializes the population, which finds out the fitness factor followed by crossover and mutation. Replace the offspring with new population and do the acceptance testing. This improves the QoS requirements of the client.

Generalized Priority Task Scheduling Algorithm

Resource monitoring in cloud computing is done in three steps—discovering and filtering the resources in broad network, selecting the appropriate resource from the available ones and finally submitted the task to desired resource. In generalized priority optimal task scheduling [17], high size task is given high priority as well as the servers with high MIPS and maps the task accordingly to the virtual machine with identified id and updates the available resources. This paper shows the improved execution time as compared to round robin and first come first serve algorithms.

Agent-Based Load Balancing

In traditional load balancing, load balancers aim at scheduling the task to appropriate servers to avoid the overloading state in the system. The agent-based dynamic load balancing uses a software entity named as mobile agent which is independent software program and run on the behalf of network user. This agent covers one walk in shared pool of resources within two walks (Figs. 2, 3 and 4).

In its first walk, it gathers all the information about the status of the servers with the average calculation of the jobs, and in its second walk it analyzes the overloaded and under loaded state of the servers and transfers the load accordingly. The additional agent in dynamic load balancing improves the throughput and response time of the system [18].

A. Estimated Finish Time Task Scheduling.

There are several computing factors like throughput, processing time, finish time, response time, etc., that counts the efficient load balancing. Faster the task execution takes place more, efficient will be the strategy. In estimated finish time task scheduling [19], the characteristics of the task are judged during the allocation and processing time in order to avoid the blocking of processes in the queue.ss It estimates its finish time at earlier stage during allocation and guides it to the appropriate server that improves the performance and resource utilization as it ensures the maximum usage of virtual machines.

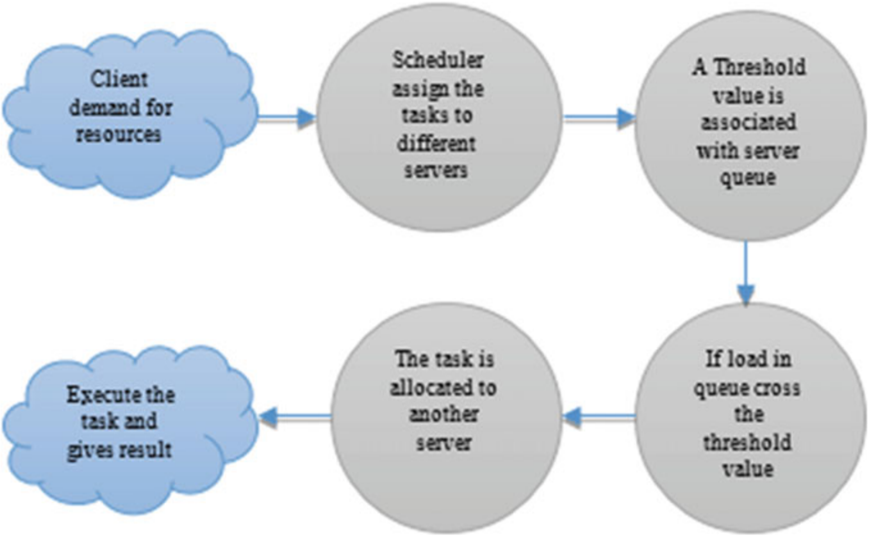
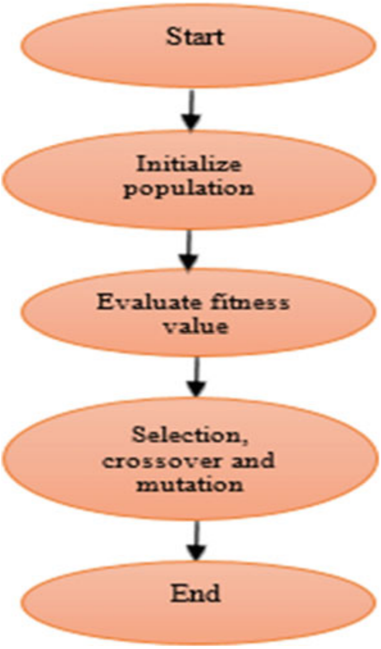

Client demand for resources
Scheduler assign the tasks to different servers
A Threshold value is associated with server queue
If load in queue cross the threshold value
The task is allocated to another server
Execute the task and gives result

Fig. 2 Process diagram of artificial honeybee

Fig. 3 Flowchart of genetic algorithm



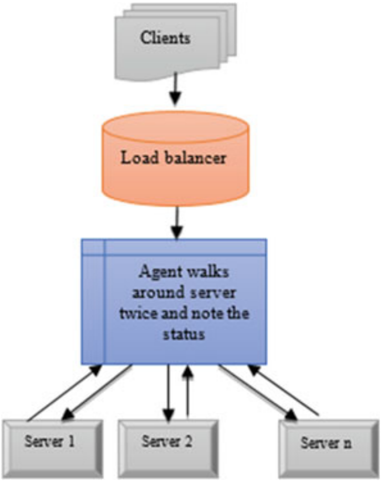


Fig. 4 Agent-based load balancing

2 Comparison and Important Findings

In this paper, comparison among several load balancing techniques has been discussed and summarized according to their performances and results. This section illustrates some important findings and performs the comparison between those existing strategies based on some measurement parameters that are listed and tabulated below:

I. Throttled and agent-based load balancing algorithm can be applied on various cost-oriented models and plays an important role in business-oriented applications and government sectors.

II. Honeybee and generalized priority focus on improving the overall execution of the system and balance the load among various nodes more efficiently as they are predictive in nature and pre-analyze the data to be allocated at different locations.

III. Resources are the most valuable assets of the computing environment and are needed to be utilized effectively so that it can contribute to the scalability and meritorious performance of the system. Ant colony estimated finish time and genetic algorithm successfully achieved the commendable resource utilization and can be modeled further with other parameters.

IV. These techniques can be integrated with other modeled load balancing techniques to achieve success in various sectors like banking, medical, forecasting, etc. (Table 1).

Table 1 Analysis of various load balancing algorithms

Metrics	Throughput	Scalability	Response time	Processing time	System stability	Cost	Performance	Resource utilization
Throttled algorithm	✗	✗	✓	✓	✗	✓	✗	✗
Ant colony optimization	✗	✗	✗	✗	✗	✗	✓	✓
Honeybee foraging	✓	✓	✗	✗	✓	✗	✗	✗
Genetic algorithm	✗	✗	✗	✗	✓	✗	✓	✓
Generalized priority	✓	✗	✗	✗	✗	✗	✗	✓
Agent-based scheduling	✓	✗	✓	✗	✗	✓	✗	✗
Estimated finish time	✗	✗	✗	✗	✗	✗	✓	✓

3 Conclusion

Cloud computing is scalable Internet-based service that aims to improve the utility of computing resources with the increase in velocity, volume and variability of incoming data. Rapid rise of data makes it difficult to handle such large amount of data and introduced a new challenge of load balancing in distributive cloud environment. Several load balancing algorithms have been proposed by intellectual researchers to efficiently direct the tasks to the computing nodes for smooth and uniform execution. Considering various proposed algorithms, this paper performs a comparative analysis among them based on different metrics. This analysis concludes that different algorithms work on different parameters, and none of them works considering all the parameters. All the proposed algorithms are efficient in one way or the other but do not claim to be the best. Therefore, these algorithms can be carried out with some new measurement parameters and can improve the quality of distribution of data with enhanced security and privacy methods.

References

1. Lakhina U, Singh N, Elamvazuthi I, Meriaudeau F, Nallagownden P, Ramasamy G, Jangra A (2018) Threshold based load handling mechanism for multi-agent micro grid using cloud computing. In: International conference on intelligent and advanced system. Kuala Lumpur
2. Linthicum DS (2017) Connecting fog and cloud computing. IEEE Cloud Comput 18–20
3. Maenhaut P-J, Moens H, Volckaert B, Ongenae V, Turck FD (2017) Resource allocation in the cloud: from simulation to experimental validation. In: IEEE 10th International conference on cloud computing (CLOUD). California
4. Geetha P, Robin CR (2017) A comparative-study of load-cloud balancing algorithms in cloud environments. In: International conference on energy, communication, data analytics and soft computing (ICECDS). Chennai
5. Jangra A, Bala R (2013) PASA: Privacy-Aware Security Algorithm for Cloud Computing. In: Abraham A, Thampi S (eds) Intelligent Informatics. Advances in intelligent systems and computing, intelligent informatics. pp 487–498, p 11, ISBN: 9783642320620
6. Singh N, Jangra A, Lakhina U An efficient load balancing algorithm for cloud computing using dynamic cluster mechanism. 3rd international conference on computing for sustainable global development, 16th–18th March 2016, IEEE, BharatiVidyapeeth's institute of computer applications and management (BVICAM), New Delhi (INDIA), Conference ID: 37465, pp 2613–2618
7. Kumar P, BundeleDM, Somwansi D (2018) An adaptive approach for load balancing in cloud computing using mtb load balancing. In 3rd International conference and workshops on recent advances and innovations in engineering (ICRAIE). Jaipur
8. Shao U, Chen J (2016) A load balancing strategy based on data correlation in cloud computing. In: 2016 IEEE/ACM 9th international conference on utility and cloud computing (UCC). Shangai.
9. Lakhina U, Singh N, Jangra A (March 2016) An efficient load balancing algorithm for cloud computing usingdynamic cluster mechanism. 3rd international conference on computing for sustainable global development. pp 2613–2618
10. Volkova VN, Chemenkaya LV, Desyatirikova EN, Hajali M, Khodar A, Osama A (2018) Load balancing in cloud computing. In: 2018 IEEE conference of russian young researchers in electrical and electronic engineering (EIConRus), pp 387–390. Moscow

11. Rahman M, Iqbal S, Gao J (April 2014) Load balancer as a service in cloud computing. In: IEEE 8th International Symposium on Service Oriented System Engineering (SOSE), 2014. pp 204–211
12. Shoja H, Nahid H, Azizi R (July 2014) A comparative survey on load balancing algorithms in cloud computing. International conference on computing, communication and networking technologies (ICCCNT), 2014. pp 1–5
13. Kumar R, Sahoo G (October 2013) Load balancing using ant colony in cloud computing. Int J InfTechnol Conver Serv (IJITCS). 3(5)
14. Dhinesh Babu LD, Krishna PV (May 2013) Honey bee behavior inspired load balancing of tasks in cloud computing environments. Appl Soft Comput 13(5):2292–2303
15. Yao J, He J-H (April 2012) Load balancing strategy of cloud computing based on artificial bee algorithm. In: 8th International conference on computing technology and information management (ICCM), 2012, vol 1. pp 185–189
16. Dasgupta K, Mandal B, Dutta P, Mandal JK, Dam S (2013) A genetic algorithm (GA) based load balancing strategy for cloud computing. First international conference on computational intelligence: modeling techniques and applications (CIMTA) 2013, vol 10. pp 340–347
17. Agarwal DA, Jain S (2014) Efficient optimal algorithm of task scheduling in cloud. Int J Comput Trends Technol (IJCTT) 9(7):344–349
18. Grover J, Katiyar S (August 2013) Agent based dynamic load balancing in cloud computing. International conference on human computer interactions (ICHCI), 2013. pp 1–6
19. Fahim Y, Lahmar EB, Labriji EH, Eddaoui A (November 2014) The load balancing based on the estimated finish time of tasks in cloud computing. Second world conference on complex systems (WCCS), 2014. pp 594–598