A Comparative Study of Load Balancing Algorithms in Cloud Computing

Damodar Tiwari[1], Shailendra Singh[2], Sanjeev Sharma[3]
[1]*Assistant Professor, CSE Department, BIST, Bhopal*
[2]*Professor, Dept of Computer Engg & Applications, NITTTR,Bhopal*
[3]*Professor & Head, SOIT, RGPV, Bhopal*

Abstract **-** Cloud computing is a high performance distributed computing efficacy which has the enormous collection of essential resources which provides basic services like the delivery of resources, platforms, and infrastructure for computing. Cloud has been provided a set of virtual computers connected to each other obtaining the maximize resource utilization, minimize the communication delay and maximize the throughput. Well-organized load balancing scheme ensures proficient resource utilization and minimum execution time by providing the resources to cloud on-demand of users' basis by equalizing the workload among all the available nodes. Load balancing can also help in reducing energy consumption and enhancing the scalability to gain maximum profits. This paper has been combining the different aspects of cloud computing and load balancing like issues, challenges, merits, demerits and performance metrics and also provided a literature review on multiple load balancing algorithms. Hence this study can help in selecting an appropriate load balancing algorithm in a cloud environment ensures the efficient resource utilization with minimum cost.

I. INTRODUCTION

Cloud computing is a subscription based on demand service delivering software, infrastructure and platform kind of services which shared resources, data, packages and alternative devices to provide cost effective, scalable and flexible utilization of heterogeneous resources [1]. It has been associated with Grid computing, Dispersed framework and Utility Computing to give good virtualization and reflection with diminishing expanse. The functioning of the cloud computing is described as selecting the tasks to the group of resources which consists a large number of computers connected by the internet and provides huge services like storage, power, and a number of software services according to the require task [20, 21, 43]. These tasks can be handled by effective load balancing algorithms to avoid overload of any single resource, improve resource consumption, reduce response time and maximize throughput. The objectives of load balancing are to retain the constancy of the system, enhance the performance, construct the fault tolerated systematic and provide a future deviation in the system such as protection, giving up customer's time and resources for additional tasks as well [23, 40].

II. CLOUD COMPUTING

Cloud Computing provides a suitable network access to shared resources such as servers, networks, storage, and services applications requiring minimum communication between service provider and management effort [43].

2.1 Cloud Deployment Models [23, 43] - Cloud Deployment Models are classified as follows:

Public Cloud: Cloud provider has been owned the public cloud services, set of policies, and costing models over the internet for general purpose consumers. Public cloud service may be free or offered as pay-per usage. Many public cloud services are Amazon EC2, S3, Google App Engine and Force.com.

Private Cloud: A single organization or company has been owned private cloud independently in which private networks are maintained services and infrastructures and provides security.

Community Cloud: Community cloud constructs and share by several organizations which contains same cloud infrastructure, their necessities and policies.

Hybrid Cloud: Hybrid cloud is an incorporated cloud service providing the combination of private, community, or public clouds together, offering the settlement of multiple exploitation models.

2.2 Cloud Service Models [23, 43] - Cloud Service Models are classified as follows:

SaaS (Software-as-a-Service): This model provides applications to cloud consumers through several interfaces such as web browsers via the internet. SaaS, cloud combines all the applications in a single environment to optimize speed, security, maintenance and availability for different cloud`s end users. Example: Gmail.

PaaS (Platform-as-a-Service): Cloud consumers hired platforms or operating systems to develop cloud services and applications which support SaaS directly on the PaaS Cloud**.** Example: Google App Engine

IaaS (Infrastructure-as-a-service): Cloud consumers use the virtual machines and physical resources in as IaaS cloud which provides IT infrastructures like networks, storage and processing. Example: Amazon's EC2.

DaaS (Data Storage as a Service): DaaS is a special case of IaaS Cloud which is used for optimizing the cost of server, post delivery services, software license and maintenance of the entire database.

Example: Google BigTable, Amazon S3, Apache and HBase etc.

2.3 Advantages of Cloud Computing [23] -

1. **On Demand Supply:** Resources are supplied on the basis of demand by users anytime anywhere.
2. **Improved Hardware Management:** All systems run on the same hardware to handle the cloud services.
3. **Catastrophe Management:** Cloud storage services provided data backup and recovery and also disaster management.
4. **Cost Effective**: IT costs such as software cost and server cost in Cloud Computing can reduce without impacting an organization's IT capabilities.
5. **Availability and Reliability**: Cloud services are available most of the time of application. Internet connection is always on and reliable for cloud end users.

III. LOAD BALANCING STRATEGIES

3.1 Cloud Computing Load Balancing Challenges [1, 21]

Cloud computing faces many challenges from the environment when it uses to choose an optimal algorithm to solve the issues of load balancing. Some of them are as follows.

Multiple Environments: Cloud computing is a combination of both heterogeneous and homogeneous environment. Both the environments have different properties, rules, regulations and criteria. So the load balancing algorithm is selected to work in both environments effectively.

Storage/Duplicacy: High storage has been required in cloud computing because same data gets stored on multiple systems which enhanced the data duplicacy and also the storage cost. Resource utilization increases the availability of the database, but also enhances the complexity of load balancing algorithms.

Complexity of Algorithm: The efficiency of the system is directly dependent on complexity of load balancing algorithms in cloud computing. More complexity of algorithms gives a negative performance in results and required the higher communication and more information.

Centralized Point of Crash: All the information in cloud is centralized at a single system or environment to provide an effective and efficient processing system, but have the issue of single point of the crash. If cloud computing used, distributed system, then the load balancing algorithm becomes more expensive and complex.

Quality of Services (Qos): All load balancing algorithm is cost efficient, accurate and flexible, which defines the quality of the algorithm in a cloud computing environment. Qos defines the qualitative and quantitative analysis of load balancing algorithm to provide the best algorithm for a special application.

3.2 Load Balancing Factors [1, 21]

Load balancing distributes the workload among the systems and provides resource utilization with superior user contentment. Load balancing ensures some characteristics which describe the usefulness of the algorithm in a cloud environment.

Throughput: it is to be described by the rate of data transfer between sender and receiver in a particular amount of time to complete a task. Higher throughput gives higher performance of the algorithm.

Response time: it is a time to be taken by a load balancing methodology for responding a submitted request.

Fault Tolerance: if any system is failing in a cloud environment, then work is transferred to another system and load is balanced by using an algorithm.

Flexibility/Scalability: the system is flexible for future modification and increased workload to maintain the efficiency and performance.

Resource Utilization: Resources are utilized by the load balancing algorithm in cloud computing to give better performance.

Performance: Performance directly depends upon the other factors which dare to be discussed. Entire factors combined to give better performance of algorithms.

3.3 Load Balancing Algorithms

There are several load balancing algorithms which are used in cloud computing to achieve better performance for applications at different level of complexities. They are also enhanced the quality of cloud computing balancing the workload between systems to ensure the better throughput, less execution time, and high processing speed [1].

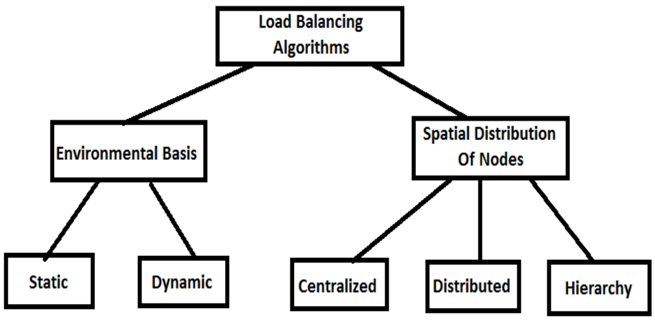

Fig1: Load Balancing Algorithms

3.3.1. Static Load Balancing - The static algorithms utilize previous knowledge about the system and resources such as storage, processing power, data according to the user's requirements and considered necessary performance and do not involve the information about the present state of the system. In case of unexpected failure of system resources and tasks these algorithms cannot be stimulated to other system in its implementation state to balance the work load. A Modified Round Robin algorithm is used to solve this problem to ensure balance traffic [1, 23, 43].

3.3.2. Dynamic Load Balancing - The dynamic algorithms consider the present state of the system during execution the load balancing. This overcomes the drawback of static algorithm. They are complex in nature to perform better and are fault tolerant in nature [1, 23, 43].

3.3.3. Centralized Load Balancing - A single central node performs all the process of scheduling and allocation of load. This central node has been keeping all the information about cloud network, load balancing and resources in both static as well as dynamic techniques. It is not reliable due to heavy load on a single node so recovery is difficult in the case of central node failure [1, 23, 43].

3.3.4. Distributed Load Balancing - Multiple nodes perform the process of load balancing, allocation and task scheduling in a local information base. It is fault tolerant due to information is available on multiple nodes so recovery is easy [1, 23, 43].

3.3.5. Hierarchical Load Balancing - Master slave method is used in hierarchical load balancing at different levels of cloud network. Tree data structure is described in the layered structure of clouds and parent node helps to make every node balanced based on gathered information [1, 23, 43].

Table1. Comparison between Different Load Balancing Algorithms

Algorithms	Familiarity	Challenges	Applications	Limitations	Types
Static	Prior information is compulsory about each node statistics	Resource & Energy Utilization, Power Consumption, Response time, Scalability	Homogenous atmosphere	Flexibility & Scalability Is not solved with changing load	Round Robin, Min-Min, Max-Min
Dynamic	Run time information of every node are required for changing load requirements	Under load & overloaded processor location, Task transfer through a remote virtual machine.	Heterogeneous atmosphere	Complicated & Time Consuming	ESCEA, Throttled, Genetic algorithm, INS (Index Name Server), Active Clustering
Centralized	Any single machine or server is liable for filling the information of complete network and updating it time to time	Threshold regulations, Throughput, Single point of Failure, Intensity Interfacing between the central server and machines in a network.	Small load network	No supported fault tolerant, central decision taking node overhead	Round Robin, Genetic Algorithm, Min-Min
Distributed	Every processor or machine of network liable for load balancing and each retain their own local databases (e.g. MIB)	Selection of machines for load balancing, Fault tolerance, Migration time, Information replaces policies, Interprocessor communication.	Huge and heterogeneous atmosphere	Complicated Algorithm, Telecommunication overhead	Reduce Active Clustering, Sampling Map, INS (Index Name Server), Biased Random
Hierarchical	Nodes or machines at multiple levels of hierarchy communicate with the machines below them to exchange network performance information	Migration time, Threshold regulations, Information replaces policies, Performance, Choice of nodes at multiple levels of network,	Average or huge size network with heterogeneous atmosphere	Low fault tolerant, Complicated	Map Reduce

3.3.6. Round Robin algorithm (RR) - The Round Robin (RR) algorithm uses the time quantum to perform the multiple tasks one at a time to utilize the principal of time scheduling. Randomly first node is selected and for a time slice a job is provided to that node so the loads are equally distributed on all the nodes in a cloud environment. The process is continuously occurring until all the nodes have been assigned the job. Round robin locally allocated requests, in which requests to be sent to the least connection nodes, so some nodes may remain idle and some nodes may be loaded heavily. Round robin gives better results if all nodes have been equally loaded and similar performance [1, 23, 43].

3.3.7. Genetic Algorithm (GA) - The Natural selection strategy mechanism is used in Genetic Algorithm (GA) to provide maximization and minimization of functions for artificial intelligence. It has been helped in balancing the loads on the basis of multiple operators like mutation, inversion, selection, etc. There are three common operations selection, genetic operation, and replacement in genetic algorithm which provided the optimal solution in distribution of loads among the nodes in cloud computing. A genetic algorithm is used the concept of genes and chromosomes to generate the best load balancing [1, 23, 43].

3.3.8. Index Name Server (INS) - The performance and efficiency of cloud storage system are improved by using access point selection optimization techniques to avoid the redundancy and storage replication of data. This system is known as Index Name Server (INS) which uses the Distributed hash table (DHT) to maintain the distribution of all the information and nodes using optimal path at a given weight and time in ad-hoc network. The shortest path is to be selected for each node in a cloud environment. Flexible system performance and resource allocation are to be noticed in INS and INS also provides the scanning of backup strategies to reduce the backup cost and enhance the accuracy [1, 23].

3.3.9. Ant Colony Optimization (ACO) - Ant Colony optimization (ACO) approach is a random search technique based on the concept of ant colonies and located the under load nodes in cloud network. Ants searches foodstuff and attach to each other through pheromone which is evaporative material on paths travelled. Ant colony based load balancing satisfies all the policies and conditions and gives better results compare to other algorithm like genetic algorithm, FCFS and hill climbing on the basis of QoS requirements. Fault Tolerant is not considered in Ant colony algorithm due to the same priority of all nodes in the cloud network [1, 23].

3.3.10. Min-Min Scheduling - It is a static algorithm in which minimum completion time of resources and execution time of unassigned tasks identify by the cloud managers. This scheduling has the prior knowledge of performance metrics about the node related to the job. Minimum time required tasks completed first then job, having maximum execution time completed last. It reduces the response time of smaller tasks having higher throughput. In this technique the larger execution time jobs wait for an unspecified period of time until all shorter jobs assigned and updated. Energy consumption and starvation problems are commonly generated by this algorithm [23, 43].

3.3.11. Central load Balancing Decision Model (CLBDM) - It is client-server based technique in which overall execution time computes working on automated administrator. If connection time achieves a defined threshold value, the present task is terminated and assigned to another node using a traditional round robin algorithm. It is a refinement of the round robin algorithm based on session switching and suffers a problem of single point failure in cloud network [1, 23].

3.3.12. Exponential Smooth forecast based on Weight Least Connection (ESWLC) - Exponential Smooth forecast based on Weight Least Connection (ESWLC) is used to handle the log connectivity based on nodes capabilities such as CPU potential, number of connections, and storage size, which assign the resource with the smallest weight to a task. It is using the historical information about nodes and their capabilities and also predict the performance of nodes for the next moment [1, 43].

3.3.13. Honey Bee Behavior (HBB) - The Honey bee behavior algorithm is one of the most popular paradigm based on behavior for food findings by honey bee to maximize the throughput in cloud computing. Bees broadcast through waggle dance after finding the location of food which provides an idea about the quality, quantity and distance of food location. According to this approach load balancing is performed in the cloud computing network in which tasks are considered as bees and food location are nodes. The overloaded nodes transfers the tasks to underloaded nodes and updated the details about load and tasks with priorities. This process increases the overall throughput and response time and reduces the waiting time in queue in cloud network [1, 43].

3.3.14. Two-Phase Scheduling Algorithm (TPSA) - Opportunistic load balancing (OLB) and Load balancing Min-Min (LBMM) are combined to give better executing efficiency in a static cloud environment. In OLB, Each task is divided into subtask and provide opportunity to allocate tasks to each and every node in a particular time interval. LBMM describes the execution time for a job and nodes having least completion period executed first. This technique suffers from the load imbalance problem which enhanced the overhead with energy consumption [1, 23, 43].

3.3.15. Stochastic Hill Climbing (SHC)

An incomplete method type algorithm performs continuously moves to uphill and stops until any neighbors have maximum value is known as Stochastic hill climbing algorithm. This algorithm consists the uphill assignments which are evaluated on some criteria closer to valid assignment. The assignments are mapped with other assignments and best element from the set is selected which will be the next assignment. This process is repeated until a solution is achieved [1].

3.3.16. Compare and Balance (CAB) - It is an intra cloud load balancing algorithm based on sampling and probability used for load balancing in a distributed environment. In CAB, the node selects randomly and compares the load itself in large distributed system. There are some necessary conditions in CAB such as execution time, local information and statelessness to reduce the migration time of nodes by transforming tasks as Red Hat cluster service [1].

3.3.17. CARTON - CARTON is a dynamic cloud control algorithm used by load balancing and Distributed Rate limiting (DRT) to achieve lower computational overhead with high performance. DRT supports the fair resource allocation and load equally transfers to the different servers. All the servers are equal in performance in dynamic environments so it provides low computation and communication overhead. It is easy to implement with less effort [1].

Table2. Performance of Different Load Balancing Algorithm

Algorithms	Nature	Response Time	Overhead	Fault Tolerance	Speed	Throughput	Future Aspects
RR	Static, Centralized	Fast	High	No	Average	High	Storage LBA over various storage processors
GA	Dynamic, centralized	Slow	High	Yes	Fast	High	It could be applied distinct crossover and selection strategies to get efficient results
INS	Dynamic	Fast	High	Yes	Fast	High	To improve response time and migration
ACO	Dynamic	Slow	High	No	Fast	High	To apply on heterogeneous environment
Min-Min Scheduling	Static	Fast	High	No	Fast	High	To apply at low and high machine, heterogeneous system
CLBDM	Dynamic	Fast	High	No	Average	High	Decentralized load balanced
ESWLC	Static	Average	High	No	Average	Average	To attain the migration time, overhead
HBB	Dynamic	Low	High	No	Average	Average	Improve it based on QoS factors and dependent tasks
TPSA	Static	Fast	High	No	Fast	High	Three level hierarchical management
SHC	Dynamic	Low	Average	No	Fast	High	Apply on real time system
CAB	Dynamic	Fast	Low	No	Average	Average	Virtual technology for live migration
CARTON	Dynamic	Fast	Average	Yes	Fast	Average	To use interfacing between TCP packets and uses control with maximum bandwidth

IV. LITERATURE REVIEW

Cloud computing is a large collection of virtual resources providing high performance distributed computing which can be accessed anywhere anytime using the network. Cloud network is used several load balancing algorithms for distribution of multiple tasks between nodes to achieve high availability of resources and enhanced the performance of cloud computing [1, 43]. Load balancing is distributed among clusters which is performed by a centralized cluster based routing protocol and artificial bee colony algorithm to enhance the throughput and quality of communication among clusters in a wireless sensor network [2]. Performance and fairness of Ad-hoc Social Networks (ASNETs) is to be enhanced by using a Biologically inspired method (BoDMaS) for Detecting and Mitigating the impact of node Selfishness. It is also used to increase the availability of resources for load balancing in cloud network [3]. Max-Min task scheduling algorithm for load balancing is updated to improve the turnaround time by dividing the tasks into two parts larger and smaller. High Mips rate resources, assigns the larger tasks and lower Mips rate resources assign smaller tasks [4]. Heuristic, metaheuristic and hyper heuristic algorithms are designed to solve several combinatorial optimization problems with lowest computational cost in Bioinformatics [5].

Adjustable and Configurable bio-inspired scheduling heuristic for cloud based systems (ACBH) is evaluated against Ant Colony Optimization (ACO) and Honey Bee Optimization (HBO) algorithm on the basis of execution time and quality of services. ACBH performs efficient task scheduling to improve and maintain the performance of cloud computing [6]. Osmosis Load Balancing (OLB) is combined with Distributed Hash Table (DHT) based on the principle of osmosis for scheduling the tasks in both homogeneous and heterogeneous cloud networks. A chord overlay mechanism is used to manage the bio inspired agents and ensure the status of cloud [7]. The combination of two bioinspired metaheuristic algorithms gives better results compare to individual algorithm in a cloud environment and mobile ad-hoc network (MANET). These algorithms show the distributed and collective behavior and better dealing with complex optimization problems. Bio inspired algorithm is also developed in a real environment with better performance [8, 9]. A new scheduler artificial bee colony algorithm is introduced in a cloud computing environment for effective load balancing using the behavior of honey bee swarms to maximize the throughput and minimize the makespan [10, 11]. A multi objective ant colony optimization algorithm for virtual machine is proposed to solve the problem of internal load balancing between physical machines. This algorithm is analyzed against ant colony optimization algorithms, first fit algorithm and a greedy algorithm [12]. Virtualization technologies help users to utilize resources for increasing the availability, hardware utilization, security and scalability. Virtual machines attain the effective scheduling for load balancing to preserve the fault tolerance level [13].

Tasks scheduling, mathematical optimization and software engineering areas largely work on a different metaheuristic algorithm to get maximum advantages over multi level cloud computing. Several real life problems efficiently solved by multiple metaheuristic load balancing algorithms in cloud network [14, 44]. Cloud task scheduling is developed for load balancing with the help of Enhanced Genetic Algorithm (EGA) solving the issues in First Come First Serve (FCFS) and Ant Colony Optimization (ACO). This NP hard task scheduling problem can be solved by EGA using the Net Beans toolkit package in the changing environment of cloud network. EGA is analyzed on the basis of execution time, QoS, availability, throughput and response time to give better results in large distributed computing, equivalent computing and gridiron computing compare to existing algorithms [15, 16, 17]. Generally the resource costs and budget costs evaluated for the entire network and multi objective optimization algorithm is selected on the basis of such details like the makespan, cost, and deadline violation rate and resource utilization using existing techniques like ant colony optimization [18]. An Energy-aware Fruit fly Optimisation Algorithm for Load Balancing (EFOA-LB) is proposed to ensure the energy conservation and effective load balancing among the virtual machines based on modern swarm intelligence in cloud network. It is inspired by foraging behavior of fruit flies and reduced the energy consumption of data center [19, 20].

There are several issues to be found in load balancing algorithms dealing with resource utilization and cost effective task scheduling using cloud environment. These algorithms ensure that they will provide economical solutions fulfilling all the conditions like minimum processing and response time, on demand services and storage management in the cloud network for a particular application [21, 23, 39]. Ant Colony Optimization (ACO) and Genetic Algorithm (GA) hybridized to combine the performance of both algorithms to overcome the drawbacks of ACO i.e. poor coverage speed and GA i.e. do not guarantee to give global optimization solution. Throttled and Equal load share algorithm based on ACO and GA is also implemented in both homogeneous and heterogeneous cloud computing environment with a Cloudsim toolkit to reduce the overall response time, data center processing time and cost [20, 31, 32]. Evolutionary algorithms have been playing a major role in cloud computing for selection of best suited virtual machines which satisfy the fitness values and generate the genetically muted and crossed over replicated fitter solutions [24]. The bioinspired algorithms can also be used to solve the problems in Wireless Sensor Network (WSN). Maximum energy consumption cost and consumption are major drawbacks in WSN which can be overcome by using cellular automata based bioinspired algorithms supporting the parallel dynamic environment. Particle Swarm Optimization (PSO) is one of the most popular load balancing algorithms used in WSN to minimize energy consumption and improve network lifetime and throughput [25, 41].

Particle Swarm Optimization (PSO) based Symbiotic Organism Search (SOS) is improved with Simulated Annealing (SA) to optimize the task scheduling, convergence rate and quality of solution in a cloud computing environment. The efficiency and performance of SASOS for synthetic and standard workload were analyzed with the help of the CloudSim toolkit under the factors like convergence speed, response time and degree of imbalance [26]. Scheduling, Cost Approach (SCA) is introduced to calculate the cost of CPU, RAM and storage by using load balancing and task scheduling algorithms among the virtual machines in a cloud environment. The results of SCA show the significant reduction in cost compared to First Come First Serve (FCFS) and Shortest Job First (SJF) algorithms under same task priority [27]. Two variants of heuristic load balancing algorithm Particle Swarm Optimization (PSO) and Genetic Algorithm (GA) are combined to GA-PSO to utilize the advantages of both i.e. high search speed, non linear and intricate engineering problems solutions in casting industries. An Improved version of GA (IGA) is also introduced in real time environment to ensure the performance in terms of response time and finish time [28, 42]. Several jobs are divided into tasks and these tasks allocated to the different nodes on demand basis by using round robin, first come, first serve and priority scheduling algorithms to improve the efficiency. An improved Round Robin algorithm is introduced in cloud computing with a new concept of assigning time quantum to individual processes depending on their priorities [29, 30, 40]. POLARPSO is proposed to enhance the behavior and performance of Particle Swarm Optimization (PSO) avoiding the problem of local minima and cloud data migration and analyzed against several popular algorithms Gray Wolf Optimizer (GWO), Sine Cosine Algorithm (SCA), MultiVerse Optimizer (MVO) and PSO. Proposed algorithm increases the capability of PSO in reaching the best solution with benchmarked functions satisfaction [33].

Honey Bee Foraging Algorithm (HFA), Active Clustering Algorithm (ACA) and Ant Colony Optimization (ACO) are merged together in distributed computing to face the challenges like server consolidation, load balancing, energy management, virtual machine migration and so on [34]. Honey Bee Foraging Algorithm (HFA) is modified and used in cloud computing to minimize the overall response time of the algorithm by migrating the load to more efficient virtual machine and virtual machine selection has been done on the basis of high throughput [35, 45]. A novel load balancing approach applies to virtual machines in which migrate the loads on the basis of resource capacity of virtual machines. It is implemented in the CloudSim simulation environment and it shows the reduction in overall energy consumption, maximize the resource utilization and profit [36]. Real world combinatorial problems like minimum spanning trees, knapsack problem, load balancing problem, and process planning problem are to be solved by Nature Inspired Meta-Heuristic algorithms and these algorithms has been increasing the quality of service and used in different applications such vehicle routing optimization, timetabling in railway optimization and load balancing in cloud computing [37, 38].

V. COMPARISON TABLE

Table3. Comparison Table

Performance Factor Authors	Used Algorithm	Throughput	Power Consumption	Resource Utilization	Space Overhead	Response Time
Ado Adamou Abba Ari [2]	Artificial Bee Colony Optimization	High	High	High	High	Low
Ahmedin Mohammed Ahmed et al. [3]	BoDMaS	Average	High	High	High	High
Akshay Kumar Gupta et al. [4]	Max-Min Task scheduling	High	Average	Low	Average	High
Ali Al Buhussain [6]	ACBH	Average	High	Average	High	Average
B. Mallikarjuna et al. [7]	Osmosis Load Balancing (OLB) combined with DHT	High	Average	High	Low	Average
M. Lawanyashri et al. [10]	Multi-objective hybrid fruit fly optimization technique	Average	Low	High	Average	High
Fatemeh Rastkhadiv et al.[11]	Artificial Bee colony algorithm	High	High	High	Average	Average
Feng Fang et al. [12]	Multi Objective ACO	High	-	Average	Average	High
G. Gayathri [13]	Modified Round Robin	High	-	Average	-	High
Harshdeep Sharma et al. [15]	Enhanced GA	High	-	High	High	Average
Hussain A Makasarwala et al. [16]	Genetic Algorithm	-	Low	Average	-	Average
Kiranveer Kaur et al. [17]	Enhanced GA	Average	-	High	High	High
Liyun Zuo et al. [18]	Multi Objective ACO	High	Average	High	Average	High
M. Lawanya Shri et al. [19]	EFOA load balancing	High	High	Average	Low	High
Mahfooz Alam et al. [21]	Efficient Load Balancing	High	Average	Average	Low	High
Mandeep Kaur et al. [22]	ACO-GA	Average	Average	High	High	Average
Markus Wagner [24]	Evolutionary Algorithm	High	Low	Average	Average	High
Michail-Antisthenis I. Tsompanas et al [25]	Cellular Automata based Bioinspired	Average	Low	High	Average	High
Mohammed Abdullahi et al [26]	PSO based SA-SOS	High	Average	Average	High	High
Mokhtar A. Alworafi et al [27]	SCA	Average	Average	-	-	Average
N. C. Brintha et al. [28]	GA-PSO	High	-	High	Average	High
Navpreet Singh et al. [29]	Round Robin & Priority based Scheduling	Average	-	-	High	Average
Negar Dordaie et al. [30]	PSO & Hill Climbing	High	-	Average	Average	Average
Pooja Yadav et al. [31]	GA-ACO	Average	Average	High	Low	Average
Rajesh Sachdeva et al. [32]	Throttled	High	-	-	Average	High
Rizik M. H. Al-Sayyed et al. [33]	POLAR PSO	High	Average	Average	Low	High
Ruchika Aggarwal et al. [34]	HFA, ACA, ACO	High	Average	High	Low	High
S. Jyothsna [35]	Modified HFA	High	High	Average	Low	Average
Sambit Kumar Mishra et al. [36]	Novel load Balancing	High	High	Average	Low	Average
Sebagenzi Jason et al. [37]	Nature Inspired	High	Average	High	Low	Average
Shabnam Sharma et al. [38]	Nature Inspired	High	Average	High	Average	High
Sukhchandan Randhawa et al. [41]	PSO	High	High	Average	Low	High
Sukhpreet Kaur et al. [42]	Improved GA	High	High	Average	Low	Average
V.V. Bhavya et al. [45]	Honey Bee Foraging algorithm	High	High	High	Average	Average

VI. CONCLUSION

Cloud computing has generally been developed for numerous application areas like business, education, medical and so on and consisted several challenges like load balancing, energy conservation, cost, virtual machine migration and time consumption. The load balancing is one of the major issues which is not solved by any single algorithm because every algorithm gives only limited performance i.e. some consider better resource utilization and maximum throughput. Some algorithm performs well with static network and some with hierarchical and some with dynamic network. So all the issues can`t be resolved by any single load balancing algorithm. This paper broadly analyzes several load balancing algorithms based on some characteristics of cloud computing and also summarizes the challenges, performance metrics, advantages and disadvantages of algorithms in tabular form. It helps in developing a new load balancing algorithm based on the existing algorithms for a particular application because existing algorithms are not completely sufficient to fulfil all the issues in cloud computing. It also helps to analyze the

efficiency of several load balancing techniques against new performance factors like scalability and fault tolerance to solve the real life problems in a cloud environment. In future an algorithm will be developed which performs on both the environment static as well as dynamic and also performs automatic load balancing in case of failure of a virtual machine. Thermal components also included in a cloud environment to save energy consumption and cost of cooling.

VII. REFERENCES

[1]. Aanjey Mani Tripathi and Sarvpal Singh, **"**A literature review on algorithms for the load balancing in cloud computing environments and their future trends", Computer Modelling & New Technologies, vol. 21 (1), pp-64-73, 2017.

[2]. Ado Adamou Abba Ari, "Bio-inspired Solutions for Optimal Management in Wireless Sensor Networks", HAL, Universite Paris Saclay, pp-1-139, 2017. (https://tel.archives-ouvertes.fr/tel-01473938)

[3]. Ahmedin Mohammed Ahmed, Xiangjie Kong, Li Liu, Feng Xia, Saeid Abolfazli, Zohreh Sanaei and Amr Tolba, "BoDMaS: Bio-inspired Selfishness Detection and Mitigation in Data Management for Ad-hoc Social Networks", Ad Hoc Networks, Elsevier, pp-1-13, 2016. (DOI: 10.1016/j.adhoc.2016.09.013)

[4]. Akshay Kumar Gupta and Pradeep S. Rawat, "Efficient Resource Utilization in Virtual Cloud Computing Environment", International Journal of Computer Applications, Volume 168, No.11, pp-1-3, 2017.

[5]. Aleksandra Swiercz, "Hyper Heuristics and Metaheuristics for Selected Bio Inspired Combinatorial Optimization Problems", Intech, pp-1-19, 2017. (http://dx.doi.org/10.5772/intechopen.69225)

[6]. Ali Al Buhussain,"Design and Analysis of an Adjustable and Configurable Bio-inspired Heuristic Scheduling Technique for Cloud Based Systems", University of Ottawa, Ottawa, Canada, pp-1-95, 2016.

[7]. B. Mallikarjuna and P. Venkata Krishna, "OLB: A Nature Inspired Approach for Load Balancing in Cloud Computing", Cybernetics And Information Technologies, Bulgarian Academy Of Sciences, Volume 15, No. 4, pp-1-11, 2015. (DOI: 10.1515/cait-2015-0060)

[8]. C. Rajan, K. Geetha, C. Rasi Priya and R. Sasikala, "Investigation on Bio-Inspired Population Based Metaheuristic Algorithms for Optimization Problems in Ad Hoc Networks", World Academy of Science, Engineering and Technology International Journal of Computer and Information Engineering, Vol. 9, No. 3, pp-1-8, 2015.

[9]. Elisabetta Di Nitto, Daniel J. Dubois and Raffaela Mirandola, "On Exploiting Decentralized Bioinspired Selforganization Algorithms to Develop Real Systems", SEAMS, pp-1-39, 2009.

[10]. M. Lawanyashri, Balamurugan Balusamy and S. Subha, "Energy-aware hybrid fruitfly optimization for load balancing in cloud environments for EHR applications", Informatics in Medicine Unlocked, Elsevier, vol. 8, pp-42–50, 2017.

[11]. Fatemeh Rastkhadiv and Kamran Zamanifar, "Task Scheduling Based On Load Balancing Using Artificial Bee Colony In Cloud Computing Environment", International Journal of Advanced Biotechnology and Research (IJBR), Vol. 7, Number 5, pp-1058-1069, 2016.

[12]. Feng Fang and Bin-Bin Qu, "Multi-objective Virtual Machine Placement for Load Balancing", ITM Web of Conferences, pp-1-9, 2017. (DOI: 10.1051/itmconf/20171101011)

[13]. G. Gayathri and R. Latha, "Implementing a Fault Tolerance Enabled Load Balancing Algorithm in the Cloud Computing Environment", IJEDR, Volume 5, Issue 1, pp-239-246, 2017.

[14]. Gurdeep Kaur and Amit Chhabra, "Evolution and Growth of Metaheuristics for Solving Variety of Problems", International Journal of Advanced Research in Computer Science (IJARCS), Volume 8, No. 5, pp-2126-2137, 2017.

[15]. Harshdeep Sharma and Gianetan Singh Sekhon, "Load Balancing in Cloud Using Enhanced Genetic Algorithm", International Journal of Innovations & Advancement in Computer Science (IJIACS), Volume 6, Issue 1, pp-13-19, 2017.

[16]. Hussain A Makasarwala and Prasun Hazari, " Using Genetic Algorithm for Load Balancing in Cloud Computing",

[17]. Kiranveer Kaur and Amritpal Kaur, "Optimal Scheduling and Load Balancing in Cloud using Enhanced Genetic Algorithm", International Journal of Computer Applications, Volume 125, No. 11, pp-1-6, 2015.

[18]. Liyun Zuo, Lei Shu, Shoubin Dong, Chunsheng Zhu and Takahiro Hara, "A Multi-Objective Optimization Scheduling Method Based on the Ant Colony Algorithm in Cloud Computing", Special Section On Big Data Services And Computational Intelligence For Industrial Systems, IEEE, pp-2687-2699, 2015. (Digital Object Identifier 10.1109/ACCESS.2015.2508940)

[19]. M. Lawanya Shri, S. Subha and Balamurugan Balusamy, "Energy-Aware Fruitfly Optimisation Algorithm for Load Balancing in Cloud Computing Environments", International Journal of Intelligent Engineering and Systems, Vol.10, No.1, pp-75-85, 2017. (DOI: 10.22266/ijies2017.0228.09)

[20]. M. Nirmala and Dr. T. Adi Lakshmi, "Energy-Efficient Load Balancing in Cloud: A Survey on Green Cloud", International Journal of Engineering Trends and Technology (IJETT), pp-192-197, 2017.

[21]. Mahfooz Alam and Zaki Ahmad Khan, "Issues and Challenges of Load Balancing Algorithm in Cloud Computing Environment", Indian Journal of Science and Technology, Vol 10 (25), pp-1-12, 2017. (DOI: 10.17485/ijst/2017/v10i25/105688,)

[22]. Mandeep Kaur and Manoj Agnihotri, "A Hybrid Technique Using Genetic Algorithm and ANT Colony Optimization for Improving in Cloud Datacenter", International Journal of Computer Sciences and Engineering (IJSE), Volume 4, Issue 8, pp-100-105, 2016.

[23]. Manpreet Kaur, "Survey on Load Balancing Algorithms in Cloud Computing", International Journal of Innovative Research in Computer and Communication Engineering, (IJIRCCE), Vol. 5, Issue 4, pp-6917-6923, 2017. (DOI: 10.15680/IJIRCCE.2017. 0504050)

[24]. Markus Wagner, "Theory and Applications of Bio-inspired Algorithms", The University of Adelaide, Adelaide, South Australia, pp-1-150, 2013.

[25]. Michail-Antisthenis I. Tsompanas, Richard Mayne, Georgios Ch. Sirakoulis, and Andrew I. Adamatzky, "A Cellular Automata Bioinspired Algorithm Designing Data Trees in Wireless Sensor Networks", Hindawi Publishing Corporation International Journal of Distributed Sensor Networks, pp-1-10, 2015, (http://dx.doi.org/10.1155/2015/471045)

[26]. Mohammed Abdullahi and Md Asri Ngadi, "Hybrid Symbiotic Organisms Search Optimization Algorithm for Scheduling of Tasks on Cloud Computing Environment", Hybrid Symbiotic Organisms Search Optimization Algorithm for Scheduling of Tasks on Cloud Computing Environment, PLoS ONE, vol. 11 (6), 2016. (doi:10.1371/journal.pone.0158229)
[27]. Mokhtar A. Alworafi, Atyaf Dhari, Asma A. Al-Hashmi, Suresha and A. Basit Darem, "Cost-Aware Task Scheduling in Cloud Computing Environment", J. Computer Network and Information Security, vol. 5, pp-52-59, 2017. (DOI: 10.5815/ijcnis.2017.05.07)
[28]. N. C. Brintha, Shajulin Benedict and J. T. Winowlin Jappes, "A Bio-Inspired Hybrid Computation for Managing and Scheduling Virtual Resources using Cloud Concepts", Appl. Math. Inf. Sci., vol. 11, No. 2, pp-565-572, 2017. (http://dx.doi.org/10.18576/amis/110228).
[29]. Navpreet Singh and Dr. Kanwalvir Singh Dhindsa, "Hybrid Scheduling Algorithm for Efficient Load Balancing In Cloud Computing", Int. J. Advanced Networking and Applications, Volume 08, Issue 05, pp- 3181-3187, 2017.
[30]. Negar Dordaie and Nima Jafari Navimipour, "A hybrid particle swarm optimization and hill climbing algorithm for task scheduling in the cloud environments", ICT Express, pp-1-6, 2017. (http://dx.doi.org/10.1016/j.icte.2017.08.001)
[31]. Pooja Yadav and Prof. Abhilasha Vyas, "Design and Implementation of Hybrid Algorithm for Load Balancing in Cloud Computing Environment", International Journal of Innovative Research in Computer and Communication Engineering, Vol. 5, Issue 6, pp-11138-11143, 2017. (DOI: 10.15680/IJIRCCE.2017. 0506020)
[32]. Rajesh Sachdeva and Sanjeev Kakkar, "A Novel Approach in Cloud Computing for Load Balancing Using Composite Algorithms", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 7, Issue 2, pp-51-56, 2017.
[33]. Rizik M. H. Al-Sayyed, Hussam N. Fakhouri, Ali Rodan and Colin Pattinson, "Polar Particle Swarm Algorithm for Solving Cloud Data Migration Optimization Problem", Modern Applied Science, Vol. 11, No. 8, pp-98-117, 2017.
[34]. Ruchika Aggarwal and Latika Gupta, "Load Balancing In Cloud Computing", IJCSMC, Vol. 6, Issue 6, pp-180-186, 2017.
[35]. S. Jyothsna, "Distributed Load Balancing in Cloud using Honey Bee Optimization", International Journal of Emerging Trends & Technology in Computer Science (IJETTCS), Volume 5, Issue 6, pp-102-106, 2016.
[36]. Sambit Kumar Mishra, Md Akram Khan, Bibhudatta Sahoo, Deepak Puthal, Mohammad S. Obaidat and KF Hsiao, "Time Efficient Dynamic Threshold-based load balancing technique for cloud computing", IEEE, pp-161-165, 2017.
[37]. Sebagenzi Jason and Suchithra R., "Nature Inspired Algorithms for Load Balancing in Cloud Computing", International Journal of Computer Applications, Volume 160, No 4, pp-7-14, 2017.
[38]. Shabnam Sharma, Dr. Ashish Kr. Luhach and Dr. Kiran Jyoti, "A Novel Approach of Load Balancing in Cloud Computing using Computational Intelligence", International Journal of Engineering and Technology (IJET), Vol. 8, No 1, pp-124-128, 2016.
[39]. Shashi Mehar and Hansa Acharya, "A Survey on Modern Load Balancing Algorithms in Cloud Computing", International Journal of Engineering Research, Volume 6, No.4, pp -228-232, 2017. (DOI : 10.5958/2319-6890.2017.00008.3)
[40]. Shweta Patel and Prof. Mayank Bhatt, "Load Balancing In Cloud Computing Using Cloudsim", International Journal of Research In Science & Engineering, Volume 3, No. 2, pp-396-401, 2017.
[41]. Sukhchandan Randhawa and Sushma Jain, "An intelligent PSO-based energy efficient load balancing multipath technique in wireless sensor networks", Turkish Journal of Electrical Engineering & Computer Sciences, vol. 25, pp-3113-3131, 2017. (doi:10.3906/elk-1606-206)
[42]. Sukhpreet Kaur and Dr. Jyotsna Sengupta, "Load Balancing using Improved Genetic Algorithm (IGA) in Cloud Computing", International Journal of Advanced Research in Computer Engineering & Technology (IJARCET), Volume 6, Issue 8, pp-1229-1233, 2017.
[43]. Suman Pandey, "Cloud Load Balancing: A Perspective Study", International Journal of Engineering And Computer Science, Volume 6, Issue 6, pp-21602-21611, 2017. (DOI: 10.18535/ijecs/v6i6.11)
[44]. Syed Hamid Hussain Madni, Muhammad Shafie Abd Latiff, Mohammed Abdullahi, Shafi'i Muhammad Abdulhamid and Mohammed Joda Usman, "Performance comparison of heuristic algorithms for task scheduling in IaaS cloud computing environment", PLoS ONE, vol. 12 (5), pp-1-26, 2017. (https://doi.org/10.1371/journal.pone.0176321)
[45]. V.V. Bhavya, K.P. Rejina and A.S. Mahesh, "An Intensification of Honey Bee Foraging Load Balancing Algorithm in Cloud Computing", International Journal of Pure and Applied Mathematics, Volume 114, No. 11, pp-127-136, 2017