

# A literature review on algorithms for the load balancing in cloud computing environments and their future trends

**Aanje Mani Tripathi\*, Sarvpal Singh**

*Madan Mohan Malviya University of Technology, Gorakhpur-273010 Uttar Pradesh, India*

*Corresponding author's e-mail: aanjejanit09@gmail.com*

*Received 15 April 2017, www.cmnt.lv*

## Abstract

Cloud computing is a computing utility which provides basic service for computing. It is a high performance distributed computing which has the huge collection of virtual resources that can be easily accessed anytime using the internet similar to pay as you go, model. A cloud defines set of virtual computers connected to each other in a form of parallel and distributed system. It ensures the dynamic provision of resources based on service level agreement (SLA) to ameliorate one or more objectives. To attain this goal several research challenges have been faced in the area of cloud computing, And the Load balancing is one of them, which aim at equalizing the workload among all the obtainable nodes by minimizing execution time, minimizing communication delays, maximising resource utilization and maximising throughput. This paper disburse a literature review of existing load balancing algorithms suggested so far and categorized under different metrics enveloping the advantages and disadvantages of each. An overview of the important research challenges of these algorithms is presented at the end with some possible ideas for improvement.

## Keywords:

Cloud computing,  
Load balancing algorithm,  
virtual machine

## 1 Introduction

Cloud computing [1] is a subscription based service like pay-as-you-go model [2] which delivers software, infrastructure and the platform kind of services [3]. These services are categorized as the Infrastructure as a service (IaaS), Platform as a service (PaaS), and Software as a service (SaaS) in the industry. Cloud computing is introduced to reduce the cost of the hardware and software. It also aims to make the next generation data center more powerful so that it can provide dynamic and flexible services to the consumer. Deployment of cloud computation makes the industry stronger and also gives the time to focus on innovation and creativity. This will lead the IT services [4] to the higher level and will help in developing the world [5].

Cloud computing is a darwinism of the parallel computing, grid computing, and distributed computing [6]). It deals with trading the resources in an efficient way according to the need of the user. Also, it is a large scale of heterogeneous resources that resides in the data centre [7]. The virtualization ability of the cloud computing hides the heterogeneity of the resources which makes it different from other computing technologies introduced previously. The other features include user-oriented approach which delivers the services as per user necessities and virtualization technology [8] that is used to pack the resources to make it scalable and flexible.

The working of the cloud computing is described as dispatching the tasks to the pool of resources which consists of several computers. It provides enormous services including storage, power, and several software services

according to the need of the task [9]). The business and virtualization technology [3] used by the cloud computing have taken the technology to a new height, leaving the responsibility of resource allocation to the virtualization of virtual machine layer. Along with these advantages, Cloud computing faces a number of research challenges such as network level migration [10], ensuring appropriate access control [11], security [12], data availability [13], Official quagmire and transitive trust issues, Data lineage, data origin and unintended leak of sensitive information [14], besides this the most frequent problem in cloud computing is load balancing. By paying more attention to the load balancing [15], like, various new features are introduced in cloud computing. Balancing of Workloads among available nodes in cloud computing is an important facet. An efficient, effective load balancing scheme ensures an efficient resource utilization [16] by the provisioning of resources [17] to cloud users on demand basis by using pay-as-you-go-scheme. Load balancing equalizes the workload among the nodes by minimizing the execution time [18], minimizing communication delays, maximizing resource utilization and maximizing the throughput.

The motive of this paper is to survey of the maximum available algorithms that have been proposed for providing a contrast to these schemes on difference metrics that examine popular load balancing algorithms with the challenges that could be addressed. Here the load balancing algorithms have been partitioned in three main categories, static, dynamic and hybrid. To the best of our knowledge & efforts, this literature survey presents load balancing algorithms with a determined focus on cloud computing.

Briefly, the contribution of this paper is as follows.

- Giving an overview of existing cloud computing load balancing challenges.
- Providing a literature review of the existing load balancing algorithms and the way of their application.
- Advantages and disadvantages of existing load balancing algorithms.
- Future research challenges of load balancing in cloud computing.

This paper examines the related work and explore the load balancing algorithms that can be categorised all the static, dynamic and hybrid algorithms. Firstly the focus is on the cloud computing load balancing challenges, description of static, dynamic and hybrid algorithms, which further continues with the discussion over several parameters on which we determine the effectiveness of algorithms. At the end a comparative analysis of these algorithms on the discussed metrics is made which would help the future researchers in their work.

Many types of research have been done in the field of cloud computing and a number of challenges identified as that counts resource provisioning, job scheduling and load balancing. In this section, we analysed some papers of load balancing in cloud computing.

In [19] author contributed to this research area by providing survey and comparative analysis on five different meta-heuristic techniques of Cloud and Grid computing : Ant colony Optimization (ACO), Genetic Algorithm (GA), Particle Swarm optimization (PSO), League Championship Algorithm (LCA) and BAT algorithm [19]. They also dispensed the comparison of these algorithms. Although, this paper only restricted to Meta- heuristic techniques.

In [20] author discussed a number of existing load balancing algorithms and dispenses the comparison on certain metrics i.e. performance, scalability and overhead etc. continued by synthesis of algorithms on certain perspective such as energy consumption and carbon emission. However this paper mainly focuses on green computing based load balancing algorithm.

Another team of authors [21] in their paper provided an overview on distributed load balancing algorithm counting parameters i.e. fault tolerance, high availability and scalability. The paper investigated three algorithms Honeybee Foraging behaviour, active clustering and biased random sampling on parameters. Though, the paper mainly focused on distributed load balancing.

In [22] author targeted at two load balancing approaches static and dynamic scheme with computational synthesis on the performance of various load balancing algorithms. It also summarized advantages and disadvantages. However, main impetus of this paper is to analyse algorithms on the basis of time factor.

In [2] author have evaluated various load balancing policies. They focused their observations on criteria including average response time, datacentre service time and total cost. The simulation results & their work prove that the round robin algorithm performance was comparatively better than other methods. The scheme presented by their

only covers limited parameters.

It is important to point out that none of the above discussed papers presents load balancing algorithms by including all three approaches static, dynamic and hybrid. Thus our work aims at including all the three approaches of load balancing algorithm with their comparative measures and covering the future challenges of each.

## 2 Load balancing strategies

### 2.1 CLOUD COMPUTING LOAD BALANCING CHALLENGES

Cloud comprises of massive resources and the management of these resources requires proper layout and high level planning. Before designing an algorithm, resource provision must be taken into consideration covering overall scenario and have to identify the main issues that could leave an impact upon the algorithm performance [17]. In this section we have discussed the challenges to be addressed while trying to propose an optimal algorithm to resolve the issues of the load balancing in cloud computing.

The Challenges to be taken into consideration are:

**Spatial Distribution of the cloud nodes:** Some algorithms are developed only for the intranet where nodes are closely located and where communication delays are avoidable. However the major challenges to develop a load balancing algorithm that could work well with the spatial distribution of the cloud nodes with the consideration of a number factors like [23]:

- Speed of network links between the nodes.
- The distance between the user and the task processing nodes.
- Distance between the nodes that involved providing the services.
- And the High Delay among spatial distributed nodes.

**Environment:** Cloud computing technology is an integration of both heterogeneous and homogeneous environment. Both environments have their own characteristics and their own differentiable criteria. For that purpose it is important to develop an efficient load balancing that works well for both environment (Mayanka Katyul 2013) (Al Nuaimi et al. 2012)

**Storage/Replication:** Any full replication algorithm could not provide an efficient utilization of the storage because, the same data gets stored at number of nodes. In case of full replication algorithm cost is the downside due to higher storage requirements. However with the partial replication algorithm we could save parts of the data space at each nodes (with a certain level of overlap) based on the capabilities like capacity and processing power of each nodes. With this capabilities it increases the resource utilization but resulted in raising the complexity of the load balancing algorithm in checking the availability of the data set parts across the different cloud nodes [23].

**Algorithm Complexity:** For an effective Load balancing algorithm Complexity should be low because layer complexity maximizes the complex operations, which may show negativity in performance issues of results or

degraded performance. Furthermore, when an algorithm requires more information and higher communication for monitoring and control, delays could prove to be troublesome and could result in efficiency drop. Therefore, load balancing algorithm must be at its simplest form (Al Nuaimi et al. 2012).

**Point of Failure:** Load balancing algorithm aims at controlling the load balance and collecting data from the different nodes. For this purpose the algorithm must be designed in such a way that it incurs no any single point of failure. Some centralized algorithms though provide an effective and efficient mechanism but have the issues of a single and central administrator for the entire system. The Distributed load balancing algorithms are more complex and require more coordination but they proved through simulation results that they are better approach and provided better solution. Hierarchical Load balancing algorithm are also a better solution as they work on master slave mode with the issues of threshold policies, information exchange criteria and failure intensity(Al Nuaimi et al. 2012)(Mayank Katyal 2013) (Al Nuaimi et al. 2012).

2.2 LOAD BALANCING METRICS

Load balancing distributes the local load among the resources and ensures resource utilization with higher user satisfaction. A suitable load balancing mechanism must have some properties that could make it distinguishable and useful. These properties should provide higher throughput, higher response time, must have fault tolerance, scalability, high performance, efficient resource utilization, and low overhead. Here we have discussed these important metrics as follows:

**Throughput:** Describes the sending and receiving rates of data of the total number of completed task on a given input at a given time unit. For better performance of cloud system high throughput rate is required. If the throughput is high then adoptability must be high [26].

**Response time:** Time taken by load balancing mechanism to respond for a submitted request [27, 28].

**Fault Tolerance:** Continue processing without stops if any node encounters a failure then the system redirect the work to another location of data. It is the capability of the mechanism [29].

**Scalability:** Scalability is a capability of the system to cope and perform under an increased or expanding workload. A system that scales well will be able to maintain or even increase its level of performance or efficiency when tested by larger operational demands [30].

**Resource Utilization:** Refers to the utilization of resources in system. An efficient load balancing algorithm must have higher resource utilization [31].

**Overhead:** Refers to the communication overhead caused by communication between the nodes during movement of tasks [29, 32].

**Performance:** Performance refers to effectiveness of the system after complete execution of load balancing algorithms. If all listed parameters perform well then it will maximize the performance of entire system [11, 33].

2.3 LOAD BALANCING ALGORITHMS

There are a number of load balancing algorithms which work to achieve their task on different layers of cloud with different level of complexities. For better load balancing researchers aim at developing more complex load balancing algorithms. But with prons it also increases its cons like processing load, overhead, and execution time (34). Load balancing algorithms can be categorized on the basis of spatial distribution of nodes (topology) and the environment shown in Figure 1. Table 1 summarizes the types of algorithms with their knowledge base information along with addressed issues and drawbacks. Table 2 classifies some existing load balancing algorithms on the basis of environment & topology.

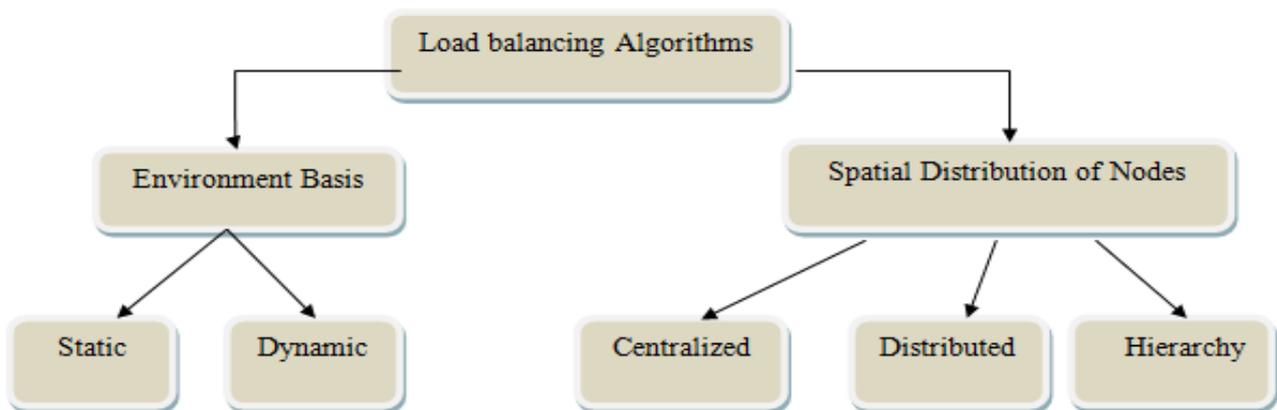


FIGURE 1 Categorizations of load balancing algorithm

TABLE 1 Types of algorithm with their knowledge base informational along with addressed issues and drawbacks

Types of Algorithms	Knowledge Base	Addressed issues	Usage	Drawbacks
Static	Previous Knowledge is mandatory about each node statistics and their user requirements	Response time Resource Utilization Scalability Power Consumption and Energy Utilization Make span	Used in Homogenous Environment	Flexibility issues Scalability issues Is not compatible with changing user requirements as well as load
Dynamic	Run time statistics of every node are observed to embrace to changing load requirements	Throughput/Performance Under loaded processor location where load will be transfer by an overloaded processor. Task transfer to a remote machine. Load Estimation. Information Gathering. Limiting the number of migration. Throughput. Threshold policies. Throughput.	Used in Heterogeneous Environment	Complexity. Time Consuming
Centralized	Any single node or server is responsible for sustaining the statistics of whole network and updating it time to time	Communication between central server and processors in network. Failure Intensity. Associated Overhead.	Useful in small network which have low load	No fault tolerant. Overhead central decision making node
Distributed	Every processor which is the part of network responsible for load balancing and each maintain their own local databases (e.g. MIB) to make efficient load balancing decisions	Selection of processor that take part in load balancing. Migration time. Interprocessor communication. Information exchange criteria. Throughput. Fault tolerance.	Useful in large and heterogeneous environment	Complexity of Algorithm. Communication overhead
Hierarchy	Nodes at different levels of hierarchy communicate with the nodes below them to get information about the network performance	Threshold policies. Information exchange criteria. Selection of nodes at different levels of network. Failure intensity. Performance. Migration time.	Useful in medium or large size network with heterogeneous environment	Less fault tolerant. Complex

TABLE 2 Classification of some existing load balancing algorithm on the basis of Environment & Topology

Dynamic	Centralized	Distributed	Hierarchy
ESCEA Throttled Biased Random Token Rating Genetic algorithm Active Clustering INS (Index Name Server)	Round Robin Min-Min Genetic Algorithm	Biased Random Sampling Map Reduce Active Clustering INS (Index Name Server)	Map Reduce

Summary: Some of the load balancing algorithms is as follows.

**Round robin algorithm**

The round robin algorithm (Dave and Maheta (2014) is one of the most popular and simplest algorithm. It allocates the resources to task or requests on the basis of time quantum. In this, time is divided into multiple slices and is allocated to the requests. It utilizes the principle of time scheduling. The resources of the service provider are provided to the requesting client on the basis of time slice. The first node is selected randomly and then it allocates job to other node on time quantum in circular manner. In round robin, loads are equally distributed on all the nodes. The scheduler begin with a node and moves on following node after a VM is assigned to that node. The iteration continued until all the nodes have been assigned to at least one VM and this process continuously occurs and is restarted from the first node. Thus, in this case, the scheduler does not need to wait for the exhaustion of the resources of a node before moving to the next node. This deficiency has been

controlled in the weighted round robin. Round robin maintains the allocation order of requests locally. It send the requests to that node which has the least number of connections, and because of this, for some periods of time, some nodes may be heavily loaded and some may remain idle [35]. This problem is solved by CLBDM (Central Load Balancing Decision Model) which is based on session switching at application layer. CLBDM calculates the connection time between user and the node and perform allocation on the basis of predefined threshold. Because of these features there has been lot of research carried out to improve performance of this algorithm. Round robin works efficiently when all the servers have the same or similar performance and are running with equal loads.

**Genetic Algorithm**

Dasgupta et al. (2013) proposed a Genetic Algorithm (GA) used as a soft computing approaches which uses the mechanism of natural selection strategy. The algorithm

balances the load of cloud infrastructure with an approach of minimizing the make span of tasks. From the simulation of this algorithm it is proven that it surpasses the existing algorithms like First Come First Serve (FCFS), local search algorithm Stochastic Hill Climbing (SHC), and Round Robin (RR). In Genetic algorithm have three operations: selection, genetic operation, and replacement. All three operations are meant for the purpose of the spread-out search space, to apply complex objective function and to avoid being trapped into local optimal solution [19, 37, 38].

#### **Index Name Server**

To avoid the redundancy and storage replication of data Wu et al. (2012) have defined a novel architecture for data centre management mechanism Index Name Server (INS), which synthesizes de-duplication with access point selection optimization techniques to upgrade the performance & efficiency of the cloud storage system. During deployment of network architecture Distributed hash table (DHT) used by Index Name Server to organize the distribution of all the data and nodes. To find an optimal path on a given weight it uses the concept of time and weight in ad-hoc network. According path preferences and to figure out the performance of each node and pick up the shortest path. To attain a flexible system performance and the best resource allocation there are several transmission matrices included in the environment and the records are table driven in INS. All the other schemes and method uses the backup strategy and it makes wastage of resources. But in INS, it excludes the scanning procedure of backup strategies and reduces the backup cost. [39] also defines future objective of INS to improve the accuracy of backup selection through considering data rates and formats, user habits, and on the basis of file formats and avoiding peak hours statistics. However, it is a centralized and complex algorithm suffers with single point failure issue (Al Nuaimi et al. 2012).

#### **Ant Colony**

Kalra et al. (2015) Joshi et al. (2014) Defined a novel Ant colony based algorithm to balance the load in cloud computing by locating the under loaded node, and experimentally proves this approach is to be more appropriate than the traditional approaches like First Come First Serve (FCFS), local search algorithm like Stochastic Hill Climbing (SHC), another soft computing approach Genetic Algorithm (GA) and some existing Ant Colony Based strategy [19]. ACO is a random search algorithm which works like ant colonies. Ants search food and connect to each other through pheromone which is evaporative stuff on paths travelled. It also guarantees that QoS requirement of Ant colony Based load balancing policy in cloud computing customer job. All the jobs are predicted to be holding the same priority though Fault torrent issues are not taken into account. Here few suggestions and ideas for the future research work are proposed on the cloud scheduling technique too [41]. The pheromone value evaluation is conducted using fault tolerance and different function variation.

#### **OLB**

For better resource utilization and improvement in response time Aditya et al. (2015) Hans et al. (2015) defined Opportunistic load balancing (OLB) ignoring expected task execution time and thus could not achieve good scheduling performance in make span. It is static load balancing

algorithm so there is no need to consider the current workload. Its aim is to keep each node involved in execution process of tasks [5]. Random execution of unexecuted task on currently available nodes is conducted. Processing of this algorithm is found to be slow because it does not calculate the current execution time.

#### **Min-Min scheduling**

Kokilavani et al. (2011) introduced Min Min Algorithm which take into account both, the minimum completion time and minimum execution time and selects nodes for executing tasks based on the Min-Min completion time. It is a static approach where the cloud manager firstly identifies the minimum execution time of unassigned task from the unassigned task set and minimum completion time of resources from all available resources. Being a static algorithm, it requires having prior knowledge of matrices related to the job. Then it assigns the task to the resource which has minimum execution time [44]. the job having maximum execution time has to wait for unspecified period of time to execute until all the tasks are assigned and updated. Results prove that this algorithm is a better scheme, that reduces the makespan than others and responds next to Genetic algorithm having rate of enhancement is also lesser in maximum scenarios [6]. Min Min algorithm also suffers with the starvation problem, and do not care about energy consumption [18].

#### **CLBDM (Central load Balancing Decision Model)**

Radojevic et al. (2011) discussed the Central load balancing decision model (CLBDM) that works as an automated administrator, compute the connection time between client and server on a given cloud resource by computing the overall execution time of task. If the connection time is over than a defined threshold then there an issue may occur (Al Nuaimi et al. 2012). And if the issue is endowed, then the task is terminated and assigned to another node using traditional round robin algorithm. CLBDM algorithm is refinement of Round Robin algorithm and is based on session switching at application layer. However, this algorithm suffered with single point failure and threshold might not be applicable in all cases.

#### **WLC**

Ren et al. (2011) introduced commonly used weighted least connection (WLC) concept that is one among the good dynamic algorithm. It does not consider parameters like distance between client and servers, service capability, processing speed, storage capacity and bandwidth. This algorithm, start with the predictions of weight of each service node and the number of connections on each service nodes. The WLC allocates the task to service node on the basis of  $\min \{ \{C(S_i)/W(S_i)\} \}$ , where C is number of allocated connections and W is the weight of service node [46]. Mean allocation is done on comparison of the sum of the connections with each service node and allocates the task having the least number of connections [47]. However, it also suffers with some issues like connections on service node cannot indicate the load well, and during long run constant weight cannot be corrected and the node is bound to divergence from the actual load condition due to which it faces load imbalance [48].

#### **ESWLC**

To handle log connectivity applications Ren et al. (2011) introduced Exponential smooth forecast based on weight least

connection (ESWLC) which allocates the resource with least weight to a task and take into account time series and tribulations. Based on the node and capabilities of the nodes, task is assigned to a node. ESWLC takes the decision to allocate a certain task to a node predicted on the basis of experience of node's cpu potency, number of connections, recollection, memory usage, the size of disk occupations. Exponential smoothing forecasting is a prediction based algorithm considering time series. This algorithm use historical data, and distinguish them using the smoothing factor. After smoothing, recent data has been makes a great impact on predictive value then long term data [48]. It establishes the training set using historical data, and then develop prediction model and the value is predicted for the next moment, which has minimum value next time continued by sending of next request connection (Al Nuaimi et al. 2012).

#### **LBMM**

Wang et al. (2010) proposed Load balancing Min Min (LBMM) using three levels of parameters for the allocation of resources in dynamic environments. This algorithm uses OLB (Opportunistic load balancing) as it base algorithm. It embraces Min Min scheduling and load balancing mechanism [18] which can avoid the non-essential assignment and utilizes the better executing efficiency. In Min Min algorithm workload of each node does not consider. It only recognizes the completion time of every task. Due to this some nodes may always get busy and some node may still remain idle. Therefore, load imbalance has been raised and the execution time of every node has been decreased. This algorithm has been processed in three layers from. In the first one the task assigned by the request manager to an appropriate service manager. The task has been divided in logical autonomous subtask by the service manager in the second phase and the execution of subtask finished in the last one. The selection of service node to execute the task has been done on the basis of the remaining CPU space (node availability), remaining memory and the transmission rate [42, 43]. However this algorithm reduces the makespan and increases the resource utilization.

#### **Biased random sampling**

[49] Defined Biased random sampling algorithm is a distributed load balancing algorithm. It creates the virtual graph which works as a knowledge base for this algorithm. And the virtual graph is a graph that represents the connection between every node and through this we know the appropriate load on the server. Every node is assumed as a vertex node and each node have a degree to represents unused resource. Each node must have one in-degree. It also uses the walk length parameter for processes, which is the traversal from one node to another. For allocating the task to a node, it begins with a random node and compares the walk length with the threshold and if it is equivalent or more than the threshold value formerly load balancer allocates the task to that node and decreases the degree of that node by one. If the degree of that node is less than one, then it is forwarded to next node which is the neighbour node of current node and walks length has been incremented by one (Randles et al. 2010). Biased random sampling algorithm performs very well with the equal or higher number of resource and provide high throughput with the utilization of increased system resources. However, it encounters performance degradation if the number of server increases

due to additional overhead to calculate the walk length.

#### **Three phase hierarchical scheduling**

Wang et al. (2011) introduced to reduce the execution time of each node, three phase hierarchical scheduling has been proposed including multiple phases of scheduling. These phases comprises BTO (Best Task Order), EOLB (Enhanced opportunistic load balancing), and EMM (Enhanced Min Min). In three phases hierarchical scheduling algorithm request monitor performs as a head of the network and is liable for observing the service manager which in turn monitors service nodes. Task execution order which is based on demand task order scheduling and service priority defined by best task order scheduling algorithm in the first phase. It stores all the tasks, subsequent tasks and tasks which are in waiting queue, in a job queue on the basis of their execution order decided by BTO. With this it reduces the waiting time and execution time of tasks. In the next phase, it uses Enhanced opportunistic load balancing algorithm that consolidate traditional opportunistic load balancing and service manager threshold. And on the basis of the job characteristics the service manager threshold allocates job on suitable node using OLB. In third phase which is Enhanced Min Min combines Min Min scheduling and service node threshold that allocates the node with the guarantee of minimum execution time taken by that node to execute (Mayanka Katyal 2013). It may be possible that EMM chooses the best service node first and then use the service node threshold to execute the task in shortest time. Three phases hierarchical scheduling algorithm confirms that jobs are executed faster and in an effective way. However this algorithm is developed under static algorithm [49].

#### **Honey Bee Behaviour**

To maximize the throughput in cloud computing paradigm Dhinesh Babu et al. (2013) have developed Honey bee behavior load balancing algorithm. This algorithm is motivated by honey bee behaviour of food findings. Bees widely search for the food and upon finding the location of food, they broadcast through waggle dance and this dance provides an idea about quality ,quantity and location as well as distance of the food. Using this idea, other bees start to acquire the food. Then again they return and perform wagle dance which provide the same ideas useful for rest of others. Same approach is applied in cloud computing for load balancing, in this when any Virtual machine has been overloaded then it migrate the task to underloaded VM, here tasks is considered as bees and food sources are VMs (Randles, Lamb, and Taleb-Bendiab 2010). After migrating the task it will update the details about load on that machine and available tasks with their priorities. This information is useful for other waiting tasks to choose VM based on their criteria as discussed. It also confirmed that a VM which has less number of high precedence task and if a high precedence task assigned to this then that task will be executed at its first. Sorting of VM will be in ascending order according to their load. This algorithm maximizes the throughput and reduces the waiting time in queue due to priority based techniques. Though here overhead is also low , but at the same time response time of VMs is found to be low.

#### **DDFTP**

Mohamed et al. (2013) defined dual direction downloading algorithm for FTP servers which provides fast

and reliable download of files. DDFTP, a dynamic algorithm which divides the file into two parts and servers start processing on the basis of a certain pattern i.e. a file  $m$  divided into  $m/2$ , then one server start downloading from the 0 zero block in incremental order (left to right) and other one start downloading from  $m$  in reverse order (right to left). Less communication between servers increases the performance and decreases the network overhead. The task is considered as accomplished when two servers start to download file on their decided patterns and new task can be assigned to servers. DDFTP also guarantees that the full utilization of communication channels i.e. if a channel bandwidth is low and other one has high then centre of file  $m$  can be change, means with high bandwidth channel download more in comparison to the low one [35]. Means  $m$  rely on the load on servers and the bandwidth of communication channels. (Al Nuaimi et al. 2012) suggested some improvements for resource utilization using partial replication whereas sustaining the similar level of performance. [48]

#### **Enhanced map reduce**

Vakil et al. (2015) defined Map Reduce which is a programming model which was implemented for the processing large data sets. Map Reduce firstly breaks the input file of job into even sized chunks and then performs replication, for the fault tolerance objective. Every single chunk developed by a map task which generates a list of key value sets. Based On the key Output of Map is split and stored in buckets. After finishing all map tasks, reduce task phase gets started which apply reduce function on map outputs corresponding to each key. Map Reduce running on a particular cluster, consisting of a master node that holds information about the data chunks. Enhanced map reduce overcome the many shortcoming of Map reduce using some other factors.

#### **HTV**

Bhatia et al. (2013) defined a method to increase the performance of data center HTV dynamic load balancing algorithm. HTV algorithm incessant examines the available resources to know the status of the node and stores in queue. Node will be sorted in the queue according their weight factor and is updated each time when persistent monitoring is done. Weight factor derived from the parameter load on the server and the response time of nodes. To allocate the resources for a new job, it will refer the queue dynamically that provides high performance and efficiency. There are some steps involved is HTV, Node information queue which stores the information regarding nodes taking into account parameters namely available space (in respect of memory and processor) and the performance of the node. HTV performance algorithm determines the load of specific node, total available space, performance and stores this information in the queue. Now, the load balancer uses these details for the proper allocation and distribution of the resources. [54] also suggested addition of some other parameters for better improvement like load on specific server and priority of user task.

#### **OLB+LBMM**

Wang et al. (2010) combined the Opportunistic load balancing (OLB) and Load balancing Min-Min for the better executing efficiency. In OLB, Each node has opportunity to execute the task and its keep busy each node. Each task is

split into subtask. LBMM considers the completion period for the job, the node having minimum completion period is executed first. However, it suffers with the load imbalance i.e. some node possess heavy load and some are idle. For this purpose OLB algorithm has been added to this algorithm. which provides better completion time along with the better resource utilization and response time [5]. Furthermore, cloud computing is not only static, it may be dynamic also. Here overhead is maximized with energy consumption also being its drawback. [20]

#### **Stochastic hill climbing**

[55] developed stochastic hill climbing which is a local search algorithm. There are two types of procedure to solve any optimization problem, first is the complete method, which provides a valid solution or prove that no such solution exists. Unfortunately, this type of algorithm requires exponential time in worst case. The other is the incomplete method that does not provide guarantee for valid solution rather than this method provides satisfying solution with high probability. However, these algorithms are most popular because of speed, effectiveness, and simplicity. Stochastic hill climbing is an incomplete method type algorithm which continuously moves to uphill and stops at 'peak' where not any neighbour have high value. Now, this algorithm chooses a random element from the uphill assignments. The probability to choose an element from that may vary with the steepness of uphill move. It starts mapping the assignment from the set of assignments and each assignment element is evaluated on some criteria which are closer to valid assignment. The best element from the set will be the next assignment [4]. This operation is repeated till the solution or to the stopping benchmark. Thus, in stochastic hill climbing algorithm have two components first one is the candidate producer used to draw one solution candidate to a set of possible successors and the second one is an evaluation measures which grades each valid solution (or invalid full assignments). Refining the evaluation leads to enhanced (or closer to valid) solutions. [55] suggested using other soft computing techniques for better improvement.

#### **Compare & Balance**

Sahu et al. (2013) defined a dynamic cost efficient compare and balance algorithm for better utilization of the resources which is based on probability to compare load of nodes/hosts. If the load of any randomly selected host has been low, then it transfer extra load on that host. It minimizes the migration time using live migration technique. A traditional load balancing algorithm only considers the memory, CPU and Bandwidth of host in any datacentre. But in dynamic compare and balance algorithm, two concepts have been used. First it is required to optimize at host level in cloud system i.e. CPU, memory and bandwidth and second, optimize the cloud system on basis of threshold which is decided on behaviour of user application. It is a green computing algorithm which tries to improve host machine efficiency by minimizing number of active host. In this, VM migration executed from the high cost to low cost physical host. It assumed that every physical host has sufficient memory. Its only disadvantage is its overhead [4, 42].

### **3 Open issues and future trends**

In this section we have discussed considerable load

balancing algorithms concerned that have not been completely and comprehensively studied till now as a research prospective. We point out issues of some algorithms and exploring these in aspects of future scope.

- Round robin algorithm work efficiently when all the servers have the same or similar performance and are running with equal loads. Performance degrades with the different load on the servers because server with minimum resources receives the next job even it has not yet been able to process the current job. Need to develop an algorithm which solves this shortcoming by utilizing novel task distribution models.
- In genetic algorithm, we can apply variation of the crossover and selection strategies as a future work for getting more efficient and tuned results.
- INS algorithm is complicated to implement and to avoid such implementation complexity need to change in the structure which makes it less complex with same performance.
- In future work of Ant colony needs to study the triggering method of ant generation and the approach for pheromone update in order to considerably minimizing the searching time for candidate nodes.
- OLB algorithm for static environment with centralised balancing and the processing of this algorithm is found to be slow because it does not calculate the current execution time. For this in the future work we have to develop an algorithm which calculates the current execution time.
- Min-Min algorithm suffers with the starvation problem, and do not care about energy consumption. However, the biggest drawback is load imbalance and which one is the central issue for cloud providers. In future work of this, have to develop an algorithm which reduces the makespan and increase the resource utilization.
- CLBDM suffers with single point failure and the threshold might not be applicable in all cases. In the future scope of this is to develop an algorithm in distributed nature with good fault tolerance.
- Three Phase Hierarchical scheduling, however this algorithm is developed under static algorithm. So, for better performance need to develop an algorithm with dynamic environment.
- DDFTP suggested some improvements for resource utilization using partial replication although maintaining the similar level of performance.

## References

- [1] Voorsluys W, Broberg J, Buyya R 2011 Introduction to Cloud Computing *Cloud Comput Princ Paradig* [Internet] (jan):1–41
- [2] Mohapatra S, Smruti Rekha K, Mohanty S 2013 A Comparison of Four Popular Heuristics for Load Balancing of Virtual Machines in Cloud Computing *International Journal of Computer Applications* p. 975–8887
- [3] Buyya R, Buyya R, Yeo CS, Yeo CS, Venugopal S, Venugopal S, et al. 2009 Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility *Futur Gener Comput Syst* [Internet] Elsevier B.V. **25**(6) 17
- [4] Sahu Y, Pateriya R K, Gupta R K 2013 Cloud server optimization with load balancing and green computing techniques using dynamic compare and balance algorithm *In Proceedings: 5th International Conference on Computational Intelligence and Communication Networks, CICN 2013* p. 527–31
- [5] Wang S C, Yan K Q, Liao W P, Wang S S 2010 Towards a load balancing in a three-level cloud computing network *In Proceedings: 2010 3rd IEEE International Conference on Computer Science and Information Technology, ICCSIT 2010* p. 108–13
- [6] Etmnani K, Naghibzadeh M 2007 *A Min-Min Max-Min selective algorithm for grid task scheduling* [Internet]. 2007 3rd IEEE/FIP International Conference in Central Asia on Internet p. 1–7
- [7] Armbrust M, Fox A, Griffith R, Joseph A, R H 2009 *Above the clouds: A Berkeley view of cloud computing* Univ California, Berkeley, Tech Rep UCB [Internet] 07–013
- [8] Lombardi F, Di Pietro R 2011 Secure virtualization for cloud computing *J Netw Comput Appl* [Internet] Elsevier **34**(4) 1113–22 Available: <http://dx.doi.org/10.1016/j.jnca.2010.06.008>
- [9] Randles M, Lamb D, Taleb-Bendiab A 2010 A comparative study into

- HTV also suggested addition of some other parameters for better improvement like load on specific server and priority of user task.
- Stochastic Hill Climbing suggested using other soft computing techniques for better improvement.

Another important concept for future research works which get low attention in current load balancing mechanisms are task migration and failure management features; hence they would have added to existing algorithm for ameliorating their efficiency.

## 4 Conclusions

Cloud computing is higher service able now a days. Thus, the load balancing turns into an enormous task that must require solving. There are many distinct mechanism suggested by the scientists and researchers to solve the threats of the load balancing and none of them any single algorithm has been addressed all the issues of load balancing. Each algorithm considers only limited issues i.e. some algorithms considers resource utilization and some high throughput. Some perform well with the static environment and some with dynamic. So, after studied a number of state of the art on load balancing algorithms it confirms that an algorithm which considers all the issues impossible to develop.

This paper widely analyzes the number of load balancing algorithms in cloud computing based on environment and spatial distribution of nodes. We also summarize the types of algorithms with their knowledge base information along with addressed issues and drawbacks. Also, we have contrived the relative scrutiny of dissimilar algorithms of load balancing with the positive factors. Meanwhile all the algorithms which we discussed are not completely sufficient; thus, need to develop a new algorithm with also consider the factors such as fault tolerance and scalability. In the further study, we can concludes the efficiency of the load balancing algorithm which affected by many parameters. Therefore, before developing any new load balancing algorithm we have to conclude many new parameters for the better performance.

## Acknowledgments

The material of this paper is based on load balancing algorithm in cloud computing and partially financially supported by TEQIP-II, Madan Mohan Malviya University of Technology, Gorakhpur, India.

- distributed load balancing algorithms for cloud computing *24th IEEE Int Conf Adv Inf Netw Appl Work WAINA* [Internet] 551–6
- [10] Zhang Q, Cheng L, Boutaba R 2010 Cloud computing: State-of-the-art and research challenges *J Internet Serv Appl*. 1(1) 7–18
- [11] Casola V, Cuomo A, Rak M, Villano U 2013 The CloudGrid approach: Security analysis and performance evaluation *Futur Gener Comput Syst*. 29(1) 387–401
- [12] Ryoo J, Rizvi S, Aiken W, Kissell J, State P 2014 *Cloud Security Auditing* (December)
- [13] Carvalho M, Cirne W, Brasileiro F, Wilkes J 2014 Long-term SLOs for reclaimed cloud computing resources *Proc ACM Symp Cloud Comput - SOCC '14* [Internet] 1–13
- [14] Haryani N, Jagli D, Sangita O, Dhanamma J, Jagli M D, Solanki R, et al. 2014 Dynamic Method for Load Balancing in Cloud Computing *Int Conf Circuits, Syst Commun Inf Technol Appl* [Internet] 5(4) 336–40
- [15] Milani A S, Navimipour N J 2016 LER URGENTE - Load balancing mechanisms and techniques in the cloud environments: Systematic literature review and future trends *J Netw Comput Appl* [Internet] Elsevier
- [16] Zhao J, Yang K, Wei X, Ding Y, Hu L, Xu G 2016 A Heuristic Clustering-Based Task Deployment Approach for Load Balancing Using Bayes Theorem in Cloud Environment *IEEE Trans Parallel Distrib Syst* [Internet] 27(2) 305–16
- [17] Zhang J, Huang H, Wang X 2016 Resource provision algorithms in cloud computing: A survey *J Netw Comput Appl* [Internet] Elsevier 64 23–42
- [18] Gopinath P P G, Vasudevan S K 2015 An In-depth analysis and study of load balancing techniques in the cloud computing environment. *Procedia Comput Sci* [Internet] Elsevier Masson SAS 50 427–32
- [19] Kalra M, Singh S 2015 A review of metaheuristic scheduling techniques in cloud computing *Egypt Informatics J* [Internet] Ministry of Higher Education and Scientific Research 16(3) 275–95
- [20] Kansal N J, Chana I 2012 Cloud Load Balancing Techniques : A Step Towards Green Computing *IJCSI Int J Comput Sci Issues* [Internet] 9(1) 238–46
- [21] Randles M, Lamb D, Taleb-Bendiab A 2010 A comparative study into distributed load balancing algorithms for cloud computing *In: 24th IEEE International Conference on Advanced Information Networking and Applications Workshops, WAINA 2010* p. 551–6
- [22] Aditya A, Chatterjee U, Gupta S 2015 A comparative study of different static and dynamic load balancing algorithm in cloud computing with special emphasis on time factor *International Journal of Current Engineering and Technology* p. 1898–907
- [23] Al Nuaimi K, Mohamed N, Al Nuaimi M, Al-Jaroodi J 2012 A survey of load balancing in cloud computing: challenges and algorithm *Proc - IEEE 2nd Symp Netw Cloud Comput Appl NCCA 2012* 137–42
- [24] Mayanka Katyal A 2013 A Comparative Study of Load Balancing Algorithms in Cloud Computing Environment *Int J Distrib Cloud Comput* [Internet] 1(2) 14
- [25] Al Nuaimi K, Mohamed N, Al Nuaimi M, Al-Jaroodi J 2012 A survey of load balancing in cloud computing: challenges and algorithms *In Proceedings - IEEE 2nd Symposium on Network Cloud Computing and Applications, NCCA 2012* p. 137–42
- [26] Patel N, Chauhan S 2015 A survey on load balancing and scheduling in cloud computing *Int J Innov Res Sci Technol*. 1(7) 185–9
- [27] Sharma A 2014 Response time based load balancing in cloud computing p. 1287–93
- [28] Fahim Y, Ben Lahmar E, Labriji E H, Eddaoui A, Ouahabi S 2015 The load balancing improvement of a data center by a hybrid algorithm in cloud computing *Colloquium in Information Science and Technology, CIST* p. 141–4
- [29] Voorsluys W, Broberg J, Venugopal S, Buyya R 2009 *Cost of virtual machine live migration in clouds: A performance evaluation* Lect Notes Comput Sci (including Subser Lect Notes Artif Intell Lect Notes Bioinformatics) 5931 LNCS:254–65
- [30] Nguyen V H, Khaddaj S, Hoppe A, Oppong E 2011 A QoS based load balancing framework for large scale elastic distributed systems. *Proceedings - 10th International Symposium on Distributed Computing and Applications to Business, Engineering and Science, DCABES 2011* p. 146–50
- [31] Sun H, Zhao T, Tang Y, Liu X 2014 A QoS-aware load balancing policy in multi-tenancy environment *Proceedings - IEEE 8th International Symposium on Service Oriented System Engineering, SOSE 2014* p. 140–7
- [32] Shao X, Jibiki M, Teranishi Y, Nishinaga N 2015 Effective Load Balancing Mechanism for Heterogeneous Range Queriable Cloud Storage *IEEE 7th Int Conf Cloud Comput Technol Sci. 2015* 405–12
- [33] Grid C 2014 Performance evaluation of load sharing policies on. *Science*
- [34] Dave S, Maheta P 2014 Utilizing round robin concept for load balancing algorithm at virtual machine level in cloud environment *Int J Comput Appl* [Internet] 94(4) 23–9
- [35] Desai T, Prajapati J 2013 A survey of various load balancing techniques and challenges in cloud computing *Int J Sci Technol Res* [Internet] 2(11) 158–61
- [36] Dasgupta K, Mandal B, Dutta P, Mandal J K, Dam S 2013 A Genetic Algorithm (GA) based load balancing strategy for cloud computing *Int Conf Comput Intell Model Tech Appl* [Internet] Elsevier B.V. 10 340–7
- [37] Zhao C, Zhang S, Liu Q, Xie J, Hu J 2009 Independent tasks scheduling based on genetic algorithm in cloud computing *5th Int Conf Wirel Commun Netw Mob Comput* [Internet] 1–4
- [38] Zha J, Wang C-D, Chen Q-L, Lu X-Y, Lai J-H 2015 Server consolidation based on hybrid genetic algorithm *Ninth Int Conf Front Comput Sci Technol* [Internet] 370–5
- [39] Wu T-Y, Lee W-T, Lin Y-S, Lin Y-S, Chan H-L, Huang J-S 2012 Dynamic load balancing mechanism based on cloud storage *Comput Commun Appl Conf (ComComAp)* 102–6
- [40] Joshi A, Goudar R H 2014 Advanced computing, networking and informatics *Smart Innov Syst Technol* [Internet] 28(2) 233–40
- [41] Shoja H, Nahid H, Azizi R 2014 A comparative survey on load balancing algorithms in cloud computing *5th Int Conf Comput Commun Netw Technol ICCCNT 2014*
- [42] Hans A, Kalra S 2015 A comprehensive study of various load balancing techniques used in cloud based biomedical services 8(2) 127–32
- [43] Kokilavani T, George Amalarethinam D I 2011 Load balanced minmin algorithm for static metatask scheduling in grid computing. *Int J Comput Appl*. 20(2) 43–9
- [44] Chen H, Wang F 2013 User-priority guided min-min scheduling algorithm for load balancing in cloud computing *Parallel Comput Technol (PARCOMPTECH) Natl Conf. 2013* 1–8
- [45] Radojevic B, Zagar M 2011 Analysis of issues with load balancing algorithms in hosted (cloud) environments *Proc 34th Int Conv MIPRO* 416–20
- [46] Ren X, Lin R, Zou H 2011 A dynamic load balancing strategy for cloud computing platform based on exponential smoothing forecast *IEEE Int Conf Cloud Comput Intell Syst*. 220–4
- [47] Lee R, Jeng B 2011 *Load-Balancing Tactics in Cloud*
- [48] Samanta P, Mondal R K 2016 Load balancing through arranging task with completion time 9(5) 273–82
- [49] Rahmeh O A, Johnson P, Taleb-Bendiab A 2008 A dynamic biased random sampling scheme for scalable and reliable grid networks. *INFOCOMP J Comput Sci* [Internet] 7(4) 1–10
- [50] Wang S C, Yan K Q, Wang S S, Chen C W 2011 A three-phases scheduling in a hierarchical cloud computing network *Proc - 3rd Int Conf Commun Mob Comput C 2011* 114–7
- [51] Dhinesh Babu L D, Venkata Krishna P 2013 Honey bee behavior inspired load balancing of tasks in cloud computing environments *Appl Soft Comput J*. 13(5) 2292–303
- [52] Mohamed N, Al-Jaroodi J, Eid A 2013 A dual-direction technique for fast file downloads with dynamic load balancing in the cloud *J Netw Comput Appl* [Internet]. Elsevier 36(4) 1116–30 Available: <http://dx.doi.org/10.1016/j.jnca.2013.01.006>
- [53] Systems D 2015 Load balancing in MapReduce on homogeneous and heterogeneous clusters: an in-depth review *Mohammad Javad Kargar and Meysam Vakili* 15 149–68
- [54] Bhatia J, Patel T, Trivedi H, Majmudar V 2013 HTV dynamic load balancing algorithm for virtual machine instances in cloud *Proc - Int Symp Cloud Serv Comput ISCOS 2012* 15–20
- [55] Mondal B, Dasgupta K, Dutta P 2012 Load balancing in cloud computing using stochastic hill climbing - a soft computing approach *Procedia Technol* [Internet] 4 783–9

- [56] Dave S, Maheta P 2014 Utilizing round robin concept for load balancing algorithm at virtual machine level in cloud environment *Int J Comput Appl.* **94**(4) 23–9
- [57] Yu Q, Chen L, Li B 2015 Ant colony optimization applied to web service compositions in cloud computing *Comput Electr Eng* [Internet] Elsevier Ltd **41** 18–27
- [58] Gao R, Wu J 2015 Dynamic load balancing strategy for cloud computing with ant colony optimization *Futur Internet* [Internet] **7**(4) 465–83
- [59] Nishant K, Sharma P, Krishna V, Gupta C, Singh K P, Rastogi R 2012 Load balancing of nodes in cloud using ant colony optimization *UKSim 14th Int Conf Comput Model Simul* [Internet] 3–8
- [60] Zhang Z, Zhang X 2010 A load balancing mechanism based on ant colony and complex network theory in open cloud computing federation *Ind Mechatronics Autom (ICIMA) 2nd Int Conf. 2010* 2 240–3

AUTHORS	
	<p><b>Aanje Mani Tripathi, Gorakhpur, India</b></p> <p><b>University studies:</b> Madan Mohan Malviya University of technology, Gorakhpur, U.P., India  <b>Scientific interest:</b> Cloud Computing, MANET, Sensor Network  <b>Publications:</b> 8  <b>Experience:</b> 4 years            He received the B.Tech degree in Information Technology from Dr. R. M. L. Awadh University, U.P. India in 2010, and the M.Tech degree in Computer Science &amp; Engineering from Madan Mohan Malviya Engg. College, U.P., India in 2013. Now he is doing Ph.D. From Madan Mohan Malviya University of Technology, U.P., India from 2014 in the area of cloud computing.</p>
	<p><b>Sarvpal Singh</b></p> <p><b>Current position, grades:</b> Dr. Sarvpal Singh is an associate professor in the Department of Computer Science &amp; Engg. Madan Mohan Malviya University of Technology, U.P., India.  <b>Scientific interest:</b> wired/wireless networks, mobile computing, cloud computing, and wireless systems  <b>Publications:</b> 18  <b>Experience:</b> 18</p>