Indian Journal of Science and Technology, Vol 10(25), DOI: 10.17485/ijst/2017/v10i25/105688, July 2017
ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

Issues and Challenges of Load Balancing Algorithm in Cloud Computing Environment

Mahfooz Alam[1*] and Zaki Ahmad Khan[2]

[1]Department of Computer Science, Al- Barkaat College of Graduate Studies, Aligarh – 202002, Uttar Pradesh, India; mahfoozalam.amu@gmail.com

[2]College of Life Science, Nanjing Agricultural University, Nanjng-Jiangsu, China; t2016041@njau.edu.cn

Abstract

Objectives: To enlighten the issues and challenges of Load Balancing Algorithm (LBA) for cloud computing environment. In this study the authors proposed a LBA for cloud computing server. **Methods/Statistical Analysis**: The numerous LBAs are compared with objective, merit, demerit and challenges of the LBA in terms of parameters such as performance, response time, scalability, overhead and many more. This study facilitates identification of efficient LBA for optimizes resource use, minimum response time, maximum throughput and avoidance of overload. **Findings:** The load balancer is permitted in most instances to provide support continuity as well as dealing with extra traffic. Consequently the useful load balancing algorithms required to find inexpensive resource usage by provisioning of resources to cloud users. The use of proposed algorithm is to improve the overhead and load imbalance factor of the system. **Application**: The Load Balancing (LB) is a vital feature of cloud computing environment. The economical LB algorithm assures cost effective resource usage by provisioning of resources to cloud users on demand schedule in pay-as-you-say-manner. The LB may actually provide assistance prioritizing users by using suitable scheduling requirements and also it is crucial part in cloud computing for maintaining the work load across various systems over the network and gives minimum processing time, minimum response time and ignores the workload.

Keywords: Cloud Computing, Load Balancing Algorithm, LEC Server, LIF, Static and Dynamic Algorithm

1. Introduction

Cloud Computing (CC) is a very broad field in IT environment. It has systems that are connected via communication network such as Internet. There are various instance of cloud computing available such as Gmail, Yahoo, Microsoft, etc. When we use the Internet connection for sending and receiving the mail so, there are many companies transfer the various services from the cloud such as Amazon, Yahoo, Microsoft Google and many more[1,2]. The advantages of cloud computing is delivering flexible, high performance and on-demand services. The cloud is completely changing our life by providing consumers with new kind of services. The consumers accept service from a cloud without paying attention to the details. The cloud computing is delivered the service over the Internet using the resources Hardware and Software. Currently, the cloud is the most useful in business, educations and other area applications. Since 2009, CC is the most popular field in computer science and engineering technology, IT industry, Amazon, Google, Yahoo and different Internet based services supplier, IBM, Microsoft

*Author for correspondence

and other IT vendors have put forward their own cloud computing technique. A variety of telecom operators are also have put a large deal of interest on CC, the extremely low cost of cloud computing platform becomes the focus of the industry. Cloud computing delivers the service through three different service approaches[1]. 1. Software-as-a-Service (SaaS) is a fast growing technique for delivering the services. It is available for users through internet, it provides various services Gmail, web based email applications, Facebook and many more, 2. Platform-as-a-Service (PaaS) is a computing platform that allows creating web applications easily and reduces cost, maintenance and complexity. It also provides better tools to software and development. In PaaS, service completely and deep understanding of user activity it gives scalability, security, reliability which is in built and 3. Infrastructure-as-a-Service (IaaS) is a growing technology. IaaS give a new model as compared to SaaS and PaaS. It gives the delivery of resource in terms of hardware, operating system, and network. It provides full scalability. IaaS is growing rapidly as compared to SaaS and PaaS. IaaS clouds are also providing additional resources like as Virtual Machine (VM), load balancers, virtual LANs, and IP addresses. The cloud community has extensively used following three cloud deployment models[3] type, 1. Private Cloud implies private means provide confidential information within an organization, institutes and build own cloud. It is used to manage and store data of their organization, 2. Public Cloud (PC) is used by singles or an organization based on their needs. Using PC, confidentiality is the major security problem. The PC is used by the common public cloud users and the cloud service giver has the complete tenure of the public cloud with its profit value, own policy, charging and costing model and 3. Hybrid Cloud is add up of at least two clouds such as private and public that rest unique entities but are limit jointly by propriety technology that allows data and application portability like as load balancing among clouds. It allows the organizations to advantage from both deployment models. It provides both cloud advantages and also good service.

Our contributions of this study are summarized as follows: a survey of the 21 existing load balancing algorithms at one place, to show the issues and challenges of Load Balancing Algorithm (LBA) in cloud computing server and the authors also present the constraints that are achieved after studying each of the algorithms. Furthermore, a proposed efficient load balancing algorithm for improving Load Imbalance Factor (LIF) and overhead.

Rest of the paper is classified as follows: In Section 2 introduces the load balancing in CC. In this section, also present the 21 standard load balancing algorithms with comparison factor in Table 1. In Section 3 description of the issues and challenges of LBA and present their constraints achieved in load balancing algorithm. In section 4, proposed the LBA and apply on Linear Extensible Cube server. In section 5 presents the experimental result. Finally in section 6 conclusion and suggestions achievable avenues for future research work.

2. Load Balancing Algorithms in Cloud Computing

The first aim of load balancing is to clearly understand the consumer requirements, the data and information can be sent and received without taking more time. LB in CC is one of the major problems without load balancing users could delays, and provide time-consuming system responses, the load can be network load, memory and CPU loads etc[4]. LB is the method of increasing the performance of Distributed System (DS). It is the method of transferring the load among different processors of DS to improve job response time and resource utilization while also ignoring a condition where few processors are overloaded while other processors are idle or doing under loaded work at any given instant of time in the system. LB calculates various terms like minimizing communication delays, minimizing execution time, maximizing throughput and maximizing resource utilization[5]. LB can be categorized as static and dynamic. Static Load Balancing (SLB) algorithm does not depend on the in progress of state of the system. It requires the prior knowledge of the system. SLB algorithm does not provide run time changes facility. It is easier but it is not well suited to heterogeneous cloud environment whereas Dynamic Load Balancing (DLB) algorithm depends on in progress of state of the system. DLB does not consider the previous state of the system. It provides better performance compare to the SLB algorithm. DLB algorithm can provide easily run time changes facility[6]. It is well suited for heterogeneous cloud environment. However, it's more correct and could give result in more efficient LB. There are numerous standard load balancing algorithms that give better results to avoid the condition of load.

Table 1. Comparison of various Load Balancing Algorithms

LBA	Objective	Merits	Demerits	Nature	Future Work
BRS	To gets LB over all nodes using RS of the system domain	To effective, reliable and scalable load balancing strategy	No capacity is handle to large load	DLB	To apply on heterogeneous system
VD	To managing hierarchal and multi-dimensional resource	To use global state information	Removes overloaded on server switch and storage	DLB	Incorporate proactive migrations based on Statistical model and historical trends
CAB	Zero-downtime reallocation and used adaptive live migration for each equilibrium solution	Very fast and decreases migration time	Let CAB be have a sufficient memory with every physical host and the migration is still slow for Open VZ VM	DLB	Xen virtual technology for live migration and to support affinity and anti affinity
CARTON	To solve optimizing problem and considerable cost for allocation of job server	For implementation is to easy and simple, computation and COM overhead is very low	COM between small UDP packets and it has no extra capacity for loss of COM packet	DLB	To use COM between TCP packets and uses control with infinite bandwidth
CLBVM	To make load balancing decisions use global state information	To improve overall performance of balances the load	It is not consider fault tolerance and load of network is constatnt	DLB	Decentralized load balanced
LBVS	To achieve replica LB module	To provides large scale net data storage and storage as a service	Use writing balancing algorithm	DLB	To apply on virtual storage for large scale
ACCLB	To balance the load for open cloud computing using ant colony and complex network	It provides long distance facility to allow shared computations and storage optimal job scheduling	It takes more time	DLB	To use the probability function for good PR to the ant colony algorithm
TPSA	To keep each node busy and attain the minimum execution time of every task using OLB, LBMM respectively	Improve performance and provide effective resource utilization	It takes more time	SLB	To manage and sustain when node is consider three level hierarchical
EDLB	It is used for real-time MMOG	It provides scaling up and down a GS on several resources according to the variable user load and decrease the cost of MMOG hosting	Occasional QoS breaches as low as 0.66% amount of load events	DLB	To maintain both cost and resources for getting the optimal balance between low-provisioning and penalties
TSALB	To provide the greatest resource utilization	To get better task response time and resource utilization	Taken more time	SLB	To apply on cloud data center
LBVM	To attain good LB and to decrease dynamic migration	To remove the issues of high migration cost and LB	Problems of more load imbalance	DLB	To improve the system PR and MT

MmLB	To decrease makespan and increase resource utilization	To use unused resources effectively for unscheduled tasks and minimum competition time	Applicable in small scale distributed system, less priority and work on small tasks	SLB	To apply low and high machine heterogeneity
DCLB	Content aware load balancing with low overhead in decentralized manner	To improve the search performance hence increasing overall performance	To narrow down search using the content information	DLB	To implement on real environment such as Apache Hadoop
SBLBID	To ignore the remote servers overloading using limit redirection rates	Without overloading decreases service response times by redirecting requests to the nearest servers	Mean response time is 29% smaller than RR and 31% smaller than SL	DLB	To achieve the overhead and migration time
LFLB	In one load balancer runs multiple LB processes	Improves overall performance of load balancer	Various process run in one load balancer	DLB	To improve migration and response time
JIQ	Fast response time at every processor without COM overhead	To efficiently decreases the system load and no COM overhead	Cannot enhance to the real response time	DLB	To decrease the complexity of JIQ algorithm
PALB	It is based on their utilization percentages decides which of the calculated nodes would be operating	To save the power and performance utilization of IaaS cloud	Performance is degraded	SLB	To energy saving and storage LBA over multiple storage processors
AC	To optimizes job assignment by connecting similar group node	To increase the resource and throughput	It apply on similar group nodes	DLB	To apply on heterogeneous system
GLBS	To reduce the make span	Generates the Qos requirement of customer job	All the jobs are the same priority which may not be actual case	DLB	For getting more efficient and tuned results could be applied variation of the crossover and selection strategies
HBLB	To achieve the maximum throughput on virtual machine	Local server action achieved through global load balancing	Throughput decreases when varieties of nodes increases	DLB	Improve HBLB by allowing for Qos factors and workflow with dependent tasks
MMLB	To sustain the TST and estimate the compilation time of the tasks	Work on larger tasks and display the current update time	Smaller task is executed in the last	SLB	Can be apply on real time system

2.1 Biased Random Sampling (BRS)

BRS algorithm uses dynamic approach in the form of virtual graph and use random sampling method to get the load balancing by efficiently using the nodes. BRS algorithm is applied regular grid network[7]. Virtual graph is constructed to provide the connectivity of every node with each node of a system. Biased method use random sampling method to achieve the LB, it provides scalable and highly reliable load balancing but performance is not good in the presence of more load. BRS method is also used to adding and deleting of the processes.

2.2 Vector Dot Load (VD)

This algorithm proposed a novel approach LB algorithm named as Vector Dot for managing the hierarchical and multi-dimensional resource constraints in such systems[8]. In this algorithm dot product play important role. It distinguishes the nodes based on their requirements with the help of this vector algorithm remove overloaded load server and switch storage.

2.3 Compare And Balance (CAB)

It's a distributed LB algorithm which is based on sampling to achieve an equilibrium solution and manage the UN balanced systems load on the basis of probability[9]. In this process the node selects randomly and compares the load itself. The implementation of CAB is intra-cloud load balancing, in decentralized method used by large distributed system. There are three necessary conditions for using this algorithm such as distributed execution, local information, simplicity and statelessness. The goal of algorithm is to reduce migration time of VM by shared storage and to achieve zero downtime reallocation of virtual machine by transforming them as Red Hat cluster service.

2.4 CARTON

This algorithm proposed a new concept for cloud control used by load balancing and Distributed Rate limiting (DRT) named as CARTON[10]. In load balancing process is equally transfer the load to different servers and used for to reduce the associated costs while DRT is guaranteed that the resource allocation is fair. The performance of all servers is equal because it considers the capacity of server for dynamic workload it provide very low Communication (COM) and computation overhead. It is effort less to implement and very simple algorithm.

2.5 Central Load Balancing Policy for Virtual Machines (CLBVM)

CLBVM algorithm was proposed for improving the overall performance of the system and load balance in distributed virtual machine[11]. The shortcoming is CLBVM algorithm is not considering the fault tolerance. In this algorithm load balancing decision takes based on globally stated information. This strategy has location rule and centralized information. The migration rule is applied partially centralized and distributed. In CLBVM algorithm, the load of all networks is constant. So, it cannot be changed frequently in any possible situation. However, each virtual machine has different naming.

2.6 Load Balancing for Virtual Storage (LBVS)

LBVS algorithm was proposed that offer a huge scale net data storage model and storage as a service model based on cloud storage[12]. LBVS used for fair share replication. LBVS two LB modules are used to achieving load balancing. In this algorithm, is achieved replica LB module using fair-share replication method. LBVS algorithm also finds robustness and enhances flexibility. It helps in getting better the utilize rate of robustness, flexibility and storage resource of the system.

2.7 Ant Colony and Complex Network Load Balancing (ACCLB)

ACCLB algorithm was proposed for finding the smallest path between a food source and the nest[13]. In ACCLB algorithm, firstly assemble all requirement of cloud processor to allocate task to the appropriate node. Once the task is started from the "head processor", the ant and the phenomenon initiates the onward movement in the pathway. To confirm whether it is an overloaded processor or not, when the ant travels in onward direction from an overloaded processor appearing for coming processor. Now, if the ant find under loaded processor still it move in onward direction in the path else for an overloaded processor it starts toward the back movement to the previous under loaded processor it found previously. ACCLB allows storage over long distances and shared computation gets optimal job scheduling. ACCLB algorithm also progress the many features related to Ant Colony algorithms such as performance of this mechanism is qualitatively analyzed and considers the characteristics of complex network.

2.8 Two-Phase Scheduling Algorithm (TPSA)

This algorithm was proposed for improve the performance and effective resource utilization[14]. It is a combination of Opportunistic LB (OLB) and LB Min-Min (LBMM). In OLB algorithm, achieve the goal of load balancing when each node in working state which depends on the LBMM. OLB schedule the next expected to be available machine in random order rather than the task expected running time on that machine. OLB keeps all nodes as busy as possible. The most important feature of OLB is its simplicity because the consideration of expected task execution time is not taken for granted, so it will generate poor make span. LBMM algorithm considers a task with respective node compares this task with other nodes to provide minimum completion time. It takes more time to complete the task because one task is divided in two number of sub task. However, LBMM algorithm decreases the scheduling length and increases the resource utilization. The two phase scheduling algorithm is presented that joint OLB and LBMM in three level cloud computing network. TPSA is used for heterogeneous system.

2.9 Event-Driven Load Balancing (EDLB)

This algorithm proposed a novel concept for cost efficient hosting of Massively Multiplayer Online Games (MMOG)[15]. In this algorithm, firstly provide that resource allocation for load and service assembles for provisioning accurate amount of resources needed for the suitable execution that assurance a well knowledge to all players. After allocation of resource for Game Session (GS) the LB service inform the game server to maintain the load. EDLB algorithm receives input as capacity events, and analysis of components of the resources and the global states of GS, generate the GSLB actions. The game session LB generates in the event driven but provides minimum applications.

2.10 Task Scheduling Algorithm based on Load Balancing (TSALB)

Task scheduling algorithm was proposed for improve response time of the task, overall performance and resource utilization of CC environment[16]. TSALB strategy cannot only meet the dynamic tasks of client's necessities but also get maximum resource utilization. Using two level task scheduling attains LB through initial mapping tasks to VMs and then VMs to host resources.

2.11 Scheduling Strategy on LB of Virtual Machine (LBVM)

LBVM was proposed that using a genetic algorithm and this algorithm achieves good LB and reduced dynamic migration of virtual machine resources[17]. It uses recent state and historical data of system needed to virtual machine resources and selects the smallest amount affective solution. It also helps in find the problem of high cost of migration and Load Imbalance Factor (LIF) thus getting better resource utilization.

2.12 Min-Min Load Balanced (MmLB)

This algorithm was proposed for minimize the scheduling length and maximize the resource utilization[18]. It works on the smaller tasks to be executed first. The Min-min Algorithm (MmA) provides minimum running time for assign the job to the node and unscheduled jobs. The MmLB algorithm has working in two modes; firstly MmA is executed then secondly, the tasks are rescheduling to draw on the unutilized resource effectively. In this algorithm, indentifies that which task having small execution and the resources create it. Thus, the task is first schedule with minimum execution time in MmA. So, in the first step executes Min-Min and the second step it selects the resources with overload and allocates them to the resource with under load.

2.13 Decentralized Content Aware LB (DCLB)

DCLB algorithm was proposed for distributed computing environments[19]. It is a new approach of content aware LB method named as workload and client aware policy. It uses a unique and special property of the requests as well as computing nodes. Decentralized manner is used to implement this strategy with small overhead. This method improves the searching performance by applying the content knowledge to narrow down the search, and hence overall performance of the system.

2.14 Server-based LB for Internet Distributed Services (SBLBID)

This algorithm proposed a new approach of LB for internet distributed services[20]. It is used for web servers which divided all over the world. Using the reduction of

the service response times under the protocol that maintain a deadline of the redirection of requisitions to the nearest remote servers without overloading them. It also uses a heuristic to help web servers to endure overloads.

2.15 Lock-Free Multiprocessing Solution for LB (LFLB)

A LFLB algorithm is attained by altering Linux kernel[21]. In this algorithm, to remove the important issue that how to send every one requisitions in a user session always to similar backend server when load balance service is operated. Multiple load balancing process is executed in one load balancer to increase the efficiency of load balancer in a multi-core environment.

2.16 Join Idle Queue (JIQ)

JIQ proposed a novel approach algorithm for distributed LB on large system[22]. Join idle Queue allows load balancing for large scale web services in to distribute load balancing. When the job arrived, there is no communication overhead occur between dispatchers and processors in JIQ algorithm. This algorithm effectively reduces the system work, but does not provide increase actual response time.

2.17 Power Aware Load Balancing (PALB)

A PALB algorithm proposed decision of the nodes which would be operating by calculating the nodes based on their utilization percentage[23]. PALB algorithm strategy to IaaS cloud systems. PALB algorithm is applied to the cluster controller of the cloud to save power and simulate request from users for virtual machine instances by using job scheduler. The aim of PALB algorithm is to sustain the accessibility to calculate nodes while decreasing the whole power consumed by the cloud architecture.

2.18 Active Clustering (AC)

This algorithm proposed a self-aggregation LB method to improve job assignments by connecting like services using local re-wiring. Active clustering load balancing works on self aggregations means it start working on the similar grouping nodes together. This clustering use the resources efficiently thereby increase the throughput of the system but performance is not good by system diversity.

2.19 Genetic Based Load Balancing Strategy (GLBS)

This algorithm proposed a new approach of load balancing strategy using genetic algorithm[24]. GLBS is trying to reduce the make span of a given tasks set on cloud infrastructure. It is also compare with other two algorithms such as first come first serve, Round Robin and stochastic Hill climbing by a local search algorithm. It is a GA algorithm used for good LB for cloud computing in three steps like initial population generation, mutation and crossover.

2.20 Honeybee Behavior Inspired LB (HB-LB)

This algorithm was proposed a nature-inspired algorithm for self-organization[25]. The objective of this algorithm is well load balance on VMs for maximizing throughput. Derivation of this algorithm is taken from the behavior of honey bees for finding their food.

As the demand maintains on changing depend upon their necessity, web servers are allocated dynamically. Grouping of servers is done under virtual server and also queued to specific virtual service queue. To compute profit each server processing request is queued to maintain the quality that honeybee possess in their waggle dance.

2.21 Max-Min Load Balanced (MMLB)

The aim of this algorithm was proposed sustain Task Status Table (TST) to evaluate the actual time workload of VMs and the estimated execution time of tasks, which can assign the workload among processors and realize the load balance[26]. The TST mainly consist of the ruining time, compilation time and current update time. In MMLB algorithm, first is select the task having maximum execution time and also calculates the estimated time of the tasks in every VM after that choose the VM with minimum compilation time and then assign the task to related VM. So, it works on the larger tasks to be executed first. It gives more priority to larger tasks and smaller jobs keep on waiting for their scheduling. Max-min algorithm is like as the MmA but the difference is larger and smaller task.

In Table 1, a comparison of aspects influencing the LBAs for cloud computing environment has been taken for the consideration and different algorithms are discussed. In this table presents comparison of all

algorithms such as objective, merits, demerits, nature of algorithms and future scope.

3. Issues and Challenges of LBA in Cloud Computing

3.1 Issues of LBA

There are numerous problems of LB algorithms. Customer satisfaction is important to distribute the work load among various nodes. Load balanced scheduling demands a number of challenging issues due to heterogeneous and dynamic nature of cloud. However, there are several merits in cloud computing such as flexibility, accessibility and cost but these merits drive the required for cloud services. As a major concern in these issues, LB allows cloud computing to "scale up to increasing demand". To achieve the desired goal load balancing provides several algorithms. Some algorithms aim at getting maximum resource utilization, minimum response time and achieve higher throughput etc. In designing any load balancing algorithm some major issues considered are[27]

3.1.1 Geographical Distributions of the Nodes

It is used in the large scaled applications like Twitter, Facebook etc. To maintain the efficiency of the system and handling fault tolerance well DS of the processors in cloud computing environment is very helpful. The geographical distribution matters a lot in the overall performance of any actual time cloud environment.

3.1.2 Static vs. Dynamic Nature of Algorithms

The issue of design of LB algorithms is based on the nature of the system which may be static algorithm and dynamic algorithm. A static algorithm works on the previous information and does not depend on recent state of the system. The sudden failure of system resource is a major drawback of this algorithm. A dynamic algorithm is better than as compared to static algorithms because it works on the recent state of the system. And there is no need of previous information of the system. In this category the algorithm is considered complex but give better fault tolerance.

3.1.3 Complexity of Algorithm

In load balancing algorithms complexity have an effect on the overall performance of the system. In terms of complexity, the algorithm is very simple but in terms of migration time, fault tolerance and response time may give poor performance. On other hand, the algorithms are difficult, but are giving better results in terms of resource utilization and throughput.

3.1.4 Traffic Analysis over Different Geographical Location

To achieve throughput and maximum resource utilization, load balancer system maintains the traffic in peak hours in every position. For any LBA is significant to analyze the traffic flow in actual time scenario over dissimilar geographic regions and accordingly to balance overall workload.

3.2 Challenges of LBA

There are numerous challenges of LB algorithms in cloud computing that require being determined before exploiting the features this technology. There are few challenges including security issues legal and compliant issues, QoS, performance, and interoperability issues[14], load balancing, data management issues[28].

Performance (PR) is very important for any computing environment for overall efficiency of the system. In load balancing algorithm, all parameter are improved then can be improve the overall system performance.

Response Time (RT) is described in distributed system how much time takes to response for particular load balancing. In the load balancing algorithms, dynamic algorithm take more response time as opposite static algorithm take shorter response time.

Scalability (SC) can be improved the system performance for a given specified amount of time.

Over Load Rejection parameter is used to improving the performance of overloaded, load balancing system. Overloaded rejection is stop when overloaded situation it associated with any LBA shows the extra cost affected in implementing algorithm. It should be as low as possible.

Communication Overhead factor comes up while load balancing of load. More communication means less will be the computation and hence speedup will be affected.

Throughput (TT) a given period of time the amount of data should be successfully moved from one place to another for provide good system performance, throughput is high.

Resource Utilization (RU) should be utilized for efficient LB in system. It is includes in automatic load balancing.

Resource utilization is less in static load balancing but dynamic load balancing give better resource utilization.
Fault Tolerant (FT) any fault can cause the system performance to decrease. So, Fault tolerant is the process to remove the unnecessary fault and give better performance of the system.

Migration Time (MT) is the time taken in movement of the jobs from one processor to other processor. Migration time is always less for giving better performance of the system.

In Table 2, a comparison on the basis of above performance metrics, the existing standard load balancing methods have been shown. In this table right tic (✓) represent the achieved constrains and wrong tic (×) represent not achieved constraints. BRS algorithm has achieved PR, SC and TT constraints whereas the same algorithm has not achieved RT, Overhead, RU, FT and MT constraints.

Table 2. Summary of challenges in Load Balancing Algorithms

LBA	PR	RT	SC	Over Head	TT	RU	FT	MT
BRS	✓	×	✓	×	✓	×	×	×
VD	×	×	×	×	×	✓	×	×
CAB	×	×	✓	✓	×	✓	×	✓
CARTON	✓	×	×	✓	×	✓	×	×
CLBVM	✓	✓	×	×	✓	✓	×	×
LBVS	✓	✓	✓	×	×	×	✓	×
ACCLB	✓	×	✓	×	×	✓	✓	×
TPSA	✓	×	×	×	×	✓	×	×
EDLB	×	×	✓	×	×	✓	×	×
TSALB	✓	✓	×	×	×	✓	×	×
LBVM	×	×	×	✓	×	✓	×	×
MmLB	✓	✓	×	✓	✓	✓	×	×
DCLB	✓	✓	✓	✓	×	✓	×	×
SBLBID	✓	✓	×	×	×	×	×	×
LFLB	✓	×	×	×	✓	×	×	×
JIQ	✓	✓	×	✓	×	×	×	×
PALB	×	✓	×	✓	✓	✓	✓	✓
AC	✓	×	✓	×	✓	×	×	×
GLBS	✓	✓	×	×	×	×	×	×
HB-LB	✓	×	✓	×	✓	×	×	×
MMLB	✓	✓	×	✓	✓	✓	×	×

In the same manner other algorithms have been shown with their achieved and unachieved constraints.

4. Proposed Work

In this section, proposed Load Balancing algorithm is named as Efficient Load Balancing (ELB) algorithm for cloud computing server. The ELB algorithm is applied on Linear Extensible Cube (LEC) server[29,30]. Basically, LEC server is a cube based multiprocessor architecture and its link functions are similar to cube based network such as ring, hypercube, folded crossed cube etc. Since, LEC server increases linearly in a cube connection shape. The six nodes LEC server is shown in Figure 1.

The LEC server has various properties as follows: the number of nodes is constant. Since the lesser the number of nodes, lesser the complexity, it will be economical. The diameter of LEC as shown lesser value comparison with other networks and degree of each is always 4. So, LEC has kept constant degree due to this its architecture is simple and economical. In LEC, the extension complexity increases in a constant manner because each extension demands single layer of 2 nodes and bisection width is equal to number of nodes. LEC network has also better capability for fault tolerance. In LEC network, if any node is faulty or failure link then it is easy to determine and can be bypassed by two additional nodes.

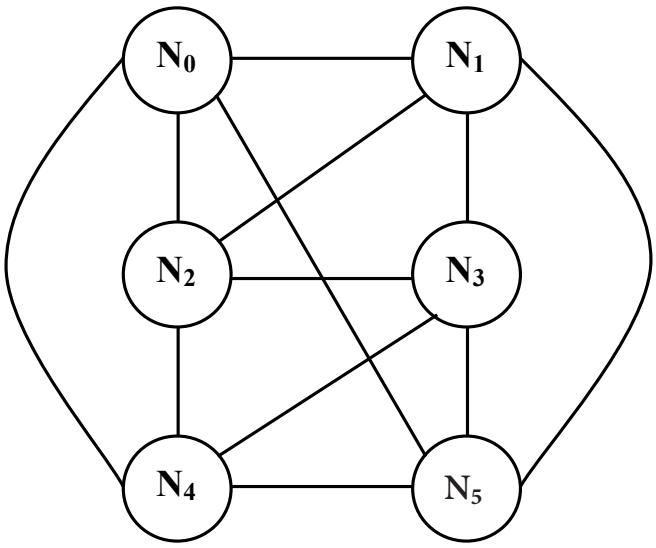

Figure 1. A simple LEC server.

Thus, LEC server is better than other type of cube based networks. So, LEC network may be economical for cloud computing server. The proposed ELB algorithm applied on LEC network for independent task scheduling is as follows:

- Task generation randomly on LEC server
- Calculate load dynamically on each node
- Implement the equal time slot selection policy at each execution node for allocation service
- ***for*** N=:0 to n ***do*** // n is number of Nodes
- Check connectivity in LEC server
  if connection exists
  {
  Migration start //Load transfer
  }
- end if
- end for
- Repeat step 5
- Repeat steps 2 - 7
- Compute LIF

5. Experimental Result

The simulation result consists of random generating artificial uniform load and mapping them on different types of cube-based interconnection networks such as Linear Extensible Cube (LEC), Hypercube (HC), Folded Hyper Cube (FHC), Crossed Cube (CC) and Cube Connected Crossed Cube (CCC). The estimation of LIF is obtained from various load stages of the independent tasks structure as shown in Table 3.

Table 3. Comparison of LIF on different cube-based interconnection networks

LEC	HC	FHC	CC	CCC
25	30	41	100	78
25	30	28	91	72
18	30	28	88	72
15	28	28	86	72
15	28	28	86	72
15	28	28	86	72
15	28	28	86	72
15	28	28	86	72
15	28	28	86	72
15	28	28	86	72

The average LIF against the load for different stages are curve plotted as Figure 2. In Figure 2, LEC represents the better LIF from other networks. The analysis of this result is shows that HC, FHC, CC and CCC have eight nodes in all networks whereas LEC network has six nodes. However, LEC has shown better LIF in this result whose we can save many resources in cloud computing environment such as energy, migration cost, power consumption, communication cost etc. It is clear from the curve that in the beginning the value of LIF starts from 25 and reaches to its maximum LIF value i.e. 15 is lesser from all the networks. Therefore, well balancing is achieved when more numbers of tasks are available. The proposed ELB algorithm is linearly extensible multiprocessor type systems in general and can be considered as low cost multiprocessor server in terms of better network utilization in cloud computing environment.

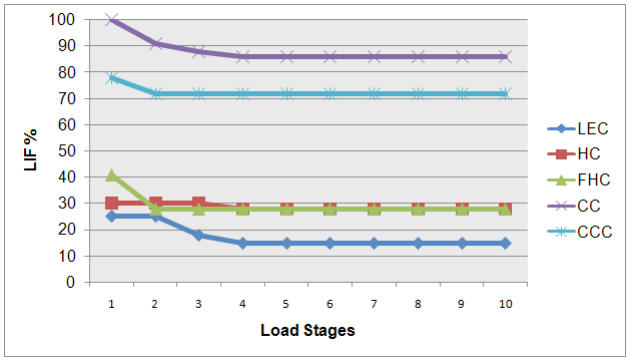

Figure 2. The LIF curve.

6. Conclusion and Future Work

The LBA is an important in the field of CC to remove the overloaded workload and provide equal and approximate service. Numerous algorithms are used to solve the complexity of load balancing. In this paper, we describe numerous standard load balancing techniques in different environments and also with different parameters. The load balancing algorithm can be used for the better utilization and understanding the requirement of the user. In this literature, we found that load balancing is an important issue on cloud system for balancing the storage, on-demand service, data center etc. It is also helps in answering the questions such as: how to achieved minimum overhead on the server and maximum resource utilization, throughput, how to reduce migration time of each processor and how to get better performance and

efficiency of system. The applications of the load balancing algorithm are a fast growing research area. Hence, on the bases of this study, proposed an ELB algorithm for cloud computing server which is to show the better performance and achieved their important constrains such as LIF, overhead, migration time. The ELB algorithm proposed has been very efficient, mainly in the case of a more number of tasks and balance of load on each node. The use of a LEC network will also efficient as it can handle all LB results with minimal inter-processor communication for cloud computing environment in future.

7. References

1. Sidhu AK, Kinger S. Analysis of load balancing techniques in cloud computing. International Journal of Computers and Technology. 2013 Mar; 4(2):737-41.
2. Reddy VK, Surya KD, Praveen MS, Lokesh B, Vishal A, Akhil K. Performance analysis of Load Balancing Algorithms in cloud computing environment. Indian Journal of Science and Technology. 2016 May; 9(18):1-7.
3. Rajeshkannan R, Aramudhan M. Comparative study of Load Balancing Algorithms in cloud computing environment. Indian Journal of Science and Technology. 2016 May; 9(20):1-7. Crossref
4. Wu TY, Lee WT, Lin YS, Lin YS, Chan HL, Huang JS. Dynamic load balancing mechanism based on cloud storage. IEEE Computing, Communications and Applications Conference (ComComAp); 2012. p. 102-6. Crossref
5. Alam M, Varshney AK. A New Approach of Dynamic Load Balancing Scheduling Algorithm for Homogeneous Multiprocessor System. International Journal of Applied Evolutionary Computation (IJAEC). 2016 Apr, 7 (2), pp. 61-75. Crossref
6. Bokhari MU, Alam M, Hasan F. Performance analysis of dynamic load balancing algorithm for multiprocessor interconnection network. Perspectives in Science (PICS). 2016 Sep; 8:564-6. Crossref
7. Rahmeh OA, Johnson P, Taleb-Bendiab A. A dynamic biased random sampling scheme for scalable and reliable grid networks. INFOCOMP Journal of Computer Science. 2008 Dec; 7(4):1-10.
8. Singh A, Korupolu M, Mohapatra D. Server-storage virtualization integration and load balancing in data centers. Proceedings of the ACM/IEEE Conference on Supercomputing; 2008 Nov. p. 10-53. Crossref
9. Zhao Y, Huang W. Adaptive distributed load balancing algorithm based on live migration of virtual machines in cloud. IEEE 5th International Joint Conference on INC, IMS and IDC, NCM'09; 2009 Aug. p. 170-5. Crossref
10. Stanojevic R, Shorten R. Load balancing vs. distributed rate limiting: An unifying framework for cloud control. IEEE International Conference on Communications; 2009 Jun. p. 1-6. Crossref
11. Chaudhary AS. Performance evaluation of web servers using central load balancing policy over virtual machines on cloud. Proceedings of the 3rd Annual ACM Bangalore Conference; 2010 Jan. p. 1-16. Crossref
12. Liu H, Liu S, Meng X, Yang C, Zhang Y. LBVS: A load balancing strategy for virtual storage. IEEE International Conference on Service Sciences; 2010 May. p. 257-62. Crossref
13. Zhang Z, Zhang X. A load balancing mechanism based on ant colony and complex network theory in open cloud computing federation. IEEE 2nd International Conference on Industrial Mechatronics and Automation (ICIMA); 2010 May. p. 240-3. Crossref
14. Wang SC, Yan KQ, Liao WP, Wang SS. Towards a load balancing in a three-level cloud computing network. 3rd IEEE International Conference Computer Science and Information Technology (ICCSIT); 2010 Jul; p. 108-13.
15. Nae V, Prodan R, Fahringer T. Cost-efficient hosting and load balancing of massively multiplayer online games. 11th IEEE/ACM International Conference on Grid Computing; 2010 Oct. p. 9-16. Crossref
16. Fang Y, Wang F, Ge J. A task scheduling algorithm based on load balancing in cloud computing. International Conference on Web Information Systems and Minin g Springer Berlin Heidelberg; 2010 Oct. p. 271-7. Crossref
17. Hu J, Gu J, Sun G, Zhao T. A scheduling strategy on load balancing of virtual machine resources in cloud computing environment. IEEE 3rd International Symposium on Parallel Architectures
18. Algorithms And Programming; 2010 Dec. p. 89-96.
19. Kokilavani T, Amalarethinam DD. Load balanced min-min algorithm for static meta-task scheduling in grid computing. International Journal of Computer Applications. 2011 Apr; 20(2):43-9.
20. Mehta H, Kanungo P, Chandwani M. Decentralized content aware load balancing algorithm for distributed computing environments. Proceedings of the International Conference and Workshop on Emerging Trends in Technology ACM; 2011 Feb. p. 370-5. Crossref
21. Nakai AM, Madeira E, Buzato LE. Load balancing for internet distributed services using limited redirection rates. IEEE 5th Latin-American Symposium on Dependable Computing (LADC); 2011 Apr. p. 156-65. https://doi.org/10.1109/ladc.2011.25
22. Liu X, Pan L, Wang CJ, Xie JY. A lock-free solution for load balancing in multi-core environment. IEEE 3rd

International Workshop on Intelligent Systems and Applications (ISA); 2011 May. p. 1-4. Crossref
23. Lu Y, Xie Q, Kliot G, Geller A, Larus JR, Greenberg A. Join-Idle-Queue A novel load balancing algorithm for dynamically scalable web services. Performance Evaluation. 2011 Nov; 68 (11):1056-71. Crossref
24. Galloway JM, Smith KL, Vrbsky SS. Power aware load balancing for cloud computing. Proceedings of the World Congress on Engineering and Computer Science; 2011 Oct. p. 19-21.
25. Dasgupta K, Mandal B, Dutta P, Mandal JK, Dam S. A genetic algorithm (ga) based load balancing strategy for cloud computing. Procedia Technology. 2013 Dec; 10:340-7. Crossref
26. Krishna PV. Honey bee behavior inspired load balancing of tasks in cloud computing environments. Applied Soft Computing. 2013 May; 13(5):2292-303. Crossref
27. Mao Y, Chen X, Li X. Max–min task scheduling algorithm for load balance in cloud computing. Proceedings of International Conference on Computer Science and Information Technology Springer; 2014. p. 457-65. Crossref
28. Khiyaita A, El Bakkali H, Zbakh M, El Kettani D. Load balancing cloud computing state of art. IEEE Network Security and Systems (JNS2); 2012 Apr. p. 106-9. Crossref
29. Buyya R, Ranjan R, Calheiros RN. Intercloud utility-oriented federation of cloud computing environments for scaling of application services. International Conference on Algorithms and Architectures for Parallel Processing Springer Berlin Heidelberg; 2010 May. p. 13-31. Crossref
30. Samad A, Rafiq MQ, Farooq O. A novel algorithm for fast retrieval of information from a multiprocessor server. Proceedings of 7th WSEAS International Conference on Software Engineering, Parallel and Distributed Systems (SEPADS'08); University of Cambridge UK. 2008 Feb. p. 68-73.
31. Khan ZA, Siddiqui J, Samad A. A novel multiprocessor architecture for massively parallel system. Proceeding of the IEEE International Conference on Parallel Distributed and Grid Computing (PDGC); India. 2014. p. 68-73. Crossref