A Methodological Survey on Load Balancing Techniques in Cloud Computing

P. Mohamed Shameem [#1] and R.S Shaji [*2]

[#]Associate Professor, Department of Computer Science and Engineering, TKM Institute of Technology, Kollam, India
pms.tkmit@yahoo.in
[*] Professor, Department of Information Technology, Noorul Islam Univeristy, Nagercoil, India
shajiswaram@yahoo.com

Abstract — **Cloud computing offers a variety of dynamic flexible resources to expedite the processing of large scale tasks in pay-by-use manner to public. In that, resource distribution and the effective load balancing among the available resources in heterogeneous environment is a major concern. Load balancing is employed across different data centers in order to establish network availability and to increase the network capacity. Moreover, the method is required to allocate dynamic workloads equally to all the nodes across cloud network. Proper load balancing technique helps in implementing fail-over, avoiding bottleneck problems, enabling scalability, optimizing resource allocation etc. This paper presents a survey about dynamic load balancing strategies directed on cloud data storage on workloads. The study also focuses on various metrics for load balancing in clouds that tends to analyze the efficacy of existing techniques. Associated overhead, response time, performance and scalability are some of the major parameters that are considered here for analysis.**

Keywords-Cloud computing, load balancing, response time, performance, resource allocation, scheduling

I.INTRODUCTION

The prodigious augmentation of cloud computing, built on the entrenched research fields of distributed computing, web services, networks, utility computing and virtualization, will afford many benefits in availability, adaptability and cost for the users. Further, on cloud computing platform, the resources are provisioned as service on demand that completely sticks to the Service Level Agreement (SLA) which has been made between the service provider and the user. Conversely, due to the requirements of the subscribers having dynamic heterogeneity and platform inconsequence, and the situation that the resources are shared; the computation will lead to wastage of resources if it cannot be evenly distributed. In addition, cloud computing platform has to balance the load effectively and dynamically among the servers that improves the resource utility and avoids hotspots.

An adept solution for managing dynamic resources on cloud platform is provided by virtualization technique which is capable of remapping resources between the virtual machines (VM) and available physical resources in accordance with load change for attaining balanced workload of the entire system in an adaptive manner. Virtual machines fully utilize the services and resources provided by the cloud to achieve better performance, since there is a highly adaptive heterogeneity of resources. Furthermore, load balancing plays a significant role in increasing resource utility in cloud [31].

In general, the cloud comprises three major components: clients, data centers and distributed servers [26]. The Figure 1 depicts the aforementioned components make up of cloud computing solution. Data center is defined here as the collection of servers hosting different applications, whereas distributed servers are the elements of a cloud that are present on internet hosting different applications. Moreover, services provided by cloud computing are categorized into 3 major types, which are as follows:

- *IaaS* (Infrastructure as a Service):

With IaaS, the components of infrastructure layer such as computation power and storage resources can be rented from the virtual resource pool for the entire industry.

- *PaaS* (Platform as a Service):

This corresponds to platform layer that made the higher level of abstraction with IaaS base. This affords the development environment, test environment, server platforms and other services.

- *SaaS* (Software as a Service):

SaaS is stated as a software distribution model, which can be accessed by the user through the internet hosting. It is necessary that the providers have to develop information for all infrastructures, software, hardware and operating systems. It is also important to offer post-maintenance and other services.

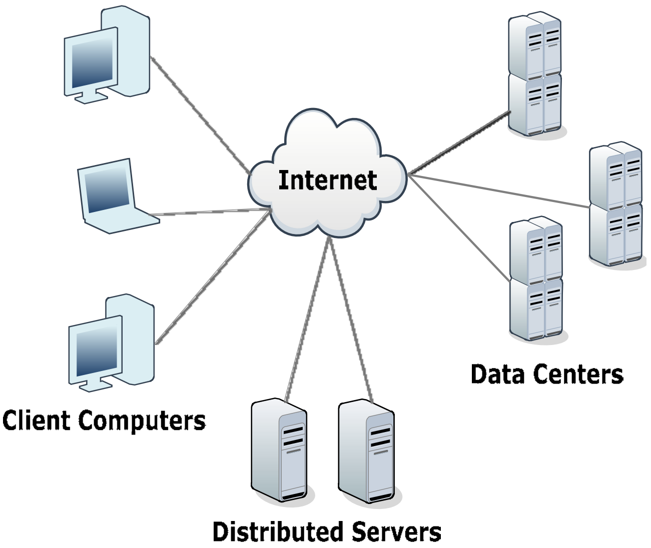

Figure 1: Components make up of Cloud Computing Solution

In cloud computing environment, load balancing is defined as the process of reassigning the total workload to the individual nodes of the composite system to facilitate effective resource utilization and to amend the response time of the job with maximum throughput. The typical load balancing algorithm is dynamic in nature, which does not concede previous state or behavior of the system; instead, it depends on the present state of the system. Load estimation, comparison of loads, system's stability, performance of the system, interaction between nodes, nature of the task to be performed and node selection are the major things that are to be considered while developing a load balancing algorithm. However, load balancing technique effectively solves the uncertainties that are happened while jobs are assigned.

The paper is organized as follows. Section 2 provides a deliberation on the load balancing techniques in distributed networks, cloud computing and also the metrics for load balancing. Section 3 presents the results and discussions. Section 4 concludes the paper with directions about future work.

II. LOAD BALANCING IN CLOUD COMPUTING : METHODS AND MATERIALS

A. Load Balancing in Distributed Networks

In distributed networks like mesh networks, P2P grid, mobile ad-hoc, wireless and overlay networks, balancing the load in an adaptive manner improves the network performance considerably. Further, the ultimate goal of load balancing is as follows:

- Even distribution of load to each resource
- Minimization of processing time for each job
- Maximum utilization of each resource

Grid serves as an extensive system for organization through which the maximum resource utilization is achieved. In [1], load balancing in grid is attained by utilizing the deduction of forward and backward ants as a competency ranks and control word to identify the appropriate resources as well. The authors performed resource power processing in grid based on tree model and ant colony model. The jobs are devoted to the resources based on its priority, node's workload and resource availability. There claimed four definitions for effective job allocation to the resource.

Definition 1: If $\forall$ job$_i$ $\exists$N resource$_j$ | ^{priority} resource$_1$= ^{priority} resource$_2$=… =^{priority} resource$_n$ $\geq$ resource$_1$, resource$_2$,…, resource$_n$ are optimal solutions.

Definition 2: If $\forall$ job$_i$ $\exists$N resource$_j$ | ^{priority} resource$_1$$\neq$ ^{priority} resource$_2$$\neq$… $\neq$^{priority} resource$_n$ $\geq$ use transitional phase.

Definition 3: If X+Y-1= X then allocate job$_i$ to resource$_j$, where i=1, 2, 3… n.

Definition 4: If X+Y-1$\neq$X then use transitional phase.

The transitional phase is the phase where provides method for the process when the site is unable to execute the user's request. The request is then given to the site which differs from the priority rank of the user's request.

The authors of [2] considered wireless mesh networks by which the available internet resources can be shared with heterogeneous capacity. There they have claimed a problem that some gateways may get overloaded rapidly. In order to overcome this problem, LARS (Load- Aware Route Selection) algorithm has been proposed. The work of this algorithm is based on queuing network model that ensures more balanced resource utilization and gateway's internet connection. The key idea of their work is to changeover the physical network model to its equivalent queuing network model for proper resource allocation. Here, the network models are represented as graphs. Moreover, the paper comprises two valuable definitions for throughput allocation and feasible throughput allocation.

Definition 1: (Throughput Allocation) A throughput allocation for G(V ∪ {a}, E_w, E_g) is any assignment for the rate λ_e and the probability matrix P_e, where E_w, E_g and λ_e are the set of edges between mesh nodes, set of edges between gateways and infrastructure and packets arrival rate respectively.

Definition 2: (Feasible Throughput Allocation) A throughput allocation is feasible for a given routing matrix R_{fwd} if every queue $q_{i,j}$(i∈ Q & I=[1, q(i)]) has a bounded time-average number of packets. This is equivalent to state that arrival process at queue $q_{i,j}$ is admissible with rate $\lambda_{i,j}$.

The outcome of LARS scheme analyzed with the routing metrics and throughput.

Further, the authors of [3] proposed an autonomous decentralized resource allocation scheme for maximizing the throughput of peer-to-peer grid. They mainly concentrated on the following three key features:

- High adaptability
- Minimized task migration
- Optimizing distributed resource discovery

The methodology combines the load rebalancing policy and dynamic estimation of arrival tasks in such a way to maximize the system throughput of peer-to-peer grid. Furthermore, the distributed node selection process for effective load balancing can be examined evenly as follows:

1. A bipartite graph with overloaded and under loaded nodes of all peers which has been divided with un-balanced loads.
2. Each overloaded peers analyses its remaining tasks and reassign those tasks to the under loaded peers in accordance with uniform selection probability.
3. Determine if each assignment is a feasible task migration function or conflict decision function.

The authors of [4] proposed an algorithm called SALB (self-acting load balancing) to tackle load imbalance in parallel file system. Specifically, they demonstrated three key characteristics of SALB.

- SALB is completely dependent on a distributed load to decision maker with distributed infrastructure.
- SALB is responsive for the network transmission.
- SALB affiliate dynamic file migration to identify its load migration.

The consideration stated that the overall system comprises three main components namely: I/O server (IOS), the metadata server (MDS) and the client. The file system interface is provided by the client that runs on computer nodes. The MDS is responsible for storing metadata such as layout data and directory information of files, whereas the IOS stores the actual data of those files. SALB makes better co-operation between IOSes to keep the load balanced. It ensures availability and scalability. Figure 2 depicts the aforementioned components are connected by a network.

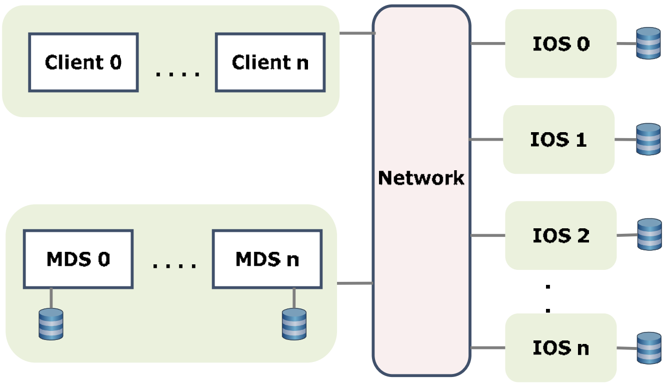

Figure 2: Architecture of Parallel File System

Concerning mobile ad-hoc networks, the authors of [5] claimed a clustering problem. They formulated the dynamic load balancing clustering problem into dynamic optimization problem. Here, they used dynamic genetic algorithms to solve the aforementioned problem by which the fitness of each individual is evaluated based on load balancing metrics. The specialized genetic algorithm for solving the problem consists of several key components such as genetic determination, fitness evaluation, population initialization, selection method, crossover and mutation. It is claimed in the paper that the clustering problem may have the characteristic that different selection sequences of cluster head will provide variant clustering results which may significantly amend the genetic algorithms exploring capability.

Fuzzy Q-learning algorithm has been proposed in [6] to resolve the load balancing problem. It is stated that fuzzy logic concedes to convert the operator experience, which is in linguistic terms into set of fuzzy rules. Moreover, fuzzy logic combined with Q-learning facilitates the task of framing fuzzy rules of the controller. Following that, fault tolerant based load balancing scheme was framed in [7], considering grid architecture, complete heterogeneity, network bandwidth, communication delay and job characteristics. The metrics they have considered for job assignment are optimal computing node utilization and minimum response time. The system model comprises grid scheduler to assign job over the grid nodes, afford resources and also receive jobs from client sites. Significantly, the paper contributes on passive replication schema for adaptive load balancing, identifying system imbalance and preventing network congestion.

Replica-aided load balancing (RALB) has been proposed in [8] for determining the weakness of overlay networks in handling time varying workloads with frequent load fluctuations. The paper also focused on designing a sophisticated cost model for estimating the load balancing cost using the factors such as load, message number and link latency. Performance tuning algorithm has also been proposed to reduce the load balancing cost. Further, the algorithm was proposed for balancing the loads among nodes in a dynamic manner in the presence of load fluctuations. The load balancing scheme proposed by the authors composed of the following steps:

Load Monitoring: Performs workload monitoring on the basis of heavily loaded and lightly loaded nodes.

Replica Placement: It is performed to enhance the reliability of services provided by the unreliable node with the account of replication cost and failure patterns.

Load Balancing: It is performed to avoid overloading among nodes while the cost is low. Moreover, the load balancing enforces two functions namely, load sharing and region reshaping.

Load Sharing: Based on the set of replicas with available capacities, workload is shared among the nodes.

Region Reshaping: It is performed to adjust the node's load assignment.

The authors have claimed that in further studies, additional parameters for tuning the network capabilities could be identified and examined.

Optimal configuration is defined as one of the best solutions for attaining effective load balancing in distributed system [9]. Ant colony algorithm is enforced for the optimized response in search space that also performs loss reduction. The following equation is considered for loss reduction in each distribution network.

$$M = \left(\frac{P_{R-loss}}{P_{I-loss}}\right) \quad (1)$$

Where P_{R-loss} and P_{I-loss} represents the network losses after and before optimal reconfiguration respectively. Further, the path selection algorithm is constructed based on the behaviour of real ants.

In order to avoid overloading on peers and optimize the network performance, the uneven load allocation problem has to be tackled effectively [11]. The authors detected the iterative key redistribution mechanism between node migration and neighbours. Here, balancing is attained by transferring key from over loaded peers to less loaded ones. Moreover, the significance for conserving order in the range- queriable data structure needs that the exchange has to be made between the neighbouring nodes. NIXMIG algorithm has been proposed here with load of each node and self imposed threshold value. The work of this algorithm has been done in three phases.

1. Exam phase
2. NIX phase
3. MIG phase

The exam phase examines the load status of all nodes to perform the suitable balancing actions. When this phase gets succeed, the nodes are locked with their workloads to proceed with the process. NIX phase is responsible for transferring the loads from one locked node to neighbour nodes. MIG phase is where the node offloading their keys to another nodes.

Churn-Resilient Protocol (CRP) has been proposed by the authors of [12] to ensure data proximity that precipitates the data distribution process under a peer-to-peer network. The work of the paper has four technical contributions, which are given as follows:

- Enhances dynamic load balancing, proximity awareness, flexible to node failures and network anomalies
- Speed gain in large data distribution is comparably high in proximity-aware overlay network than unstructured peer-to-peer networks
- The CRP based network requires less number of control messages

- CRP provides automatic broadcast of all data items

Further, the authors have proposed fast data dissemination algorithm with an effective tree construction mechanism. The churn resilience and recovery mechanism has also been imported in which each node stores the information of its ancestor nodes on the tree.

A novel load balancing algorithm [15] that is dependent on partial knowledge of the system to evaluate the dissemination probabilities and virtual server loads and imperfect knowledge of the system state. The equation for evaluating the expected load per unit capacity is given as,

$$A = \frac{\sum_{vs \in VS} l_{vs}}{\sum_{i \in N} c_i} \quad (2)$$

Where N represents the set of peers, VS states the virtual servers hosted by the peers and i stands the current load.

Load balancing in Distributed Virtual Environment (DVE) [16] is a significant task for increasing the network scalability, which can be achieved using multi server architecture. The authors focused on the quality of distributing load among the servers and the efficacy of separating workloads itself. Further, in [32], guidance was provided for dynamic load balancing in distributed systems. The author demonstrated some of the major goals of load balancing algorithm, which are given as follows:

- Improving performance at reasonable cost
- To treat all jobs equally
- Fault tolerance
- Adaptability
- System stability

Moreover, the algorithm considers the information strategy, location strategy; transfer strategy, load measurement and performance measurement for effectively formulate the dynamic load balancing algorithm.

B. Load Balancing Techniques in Cloud Computing:

As is well-known, cloud computing exploits a variety of computing resources to expedite the execution of large scale tasks. The paper [10] proposed an algorithm called LB3M (Load Balance Max-Min and Max) algorithm. The algorithm is completely based on the average completion of each task assigned to the node, node's capacity and utilization of computing resources. While concerning about the efficient load balancing in cloud, dynamic resource allocation plays a significant role. It is stated that [13], generally, cloud computing companies require parallel data processing to make the deployment of services more facile in their product portfolio. The authors have developed Nephele architecture for managing the job allocation process effectively through the cloud controller.

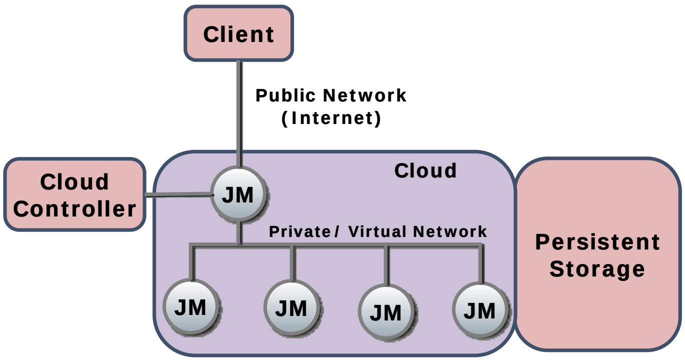

Figure 3: Structural Overview of Nephele

The architecture follows the master-slave pattern as illustrated in Figure 3. It is claimed that before enduring a task to the Nephele, the virtual machine in cloud must be started, which runs the Job Manager (JM). The JM is responsible for receiving client's job and schedule them. The cloud controller is an interface, responsible for communications by which the JM can allocate or de-allocate the job to VMs based on current job execution phase. The actual execution of tasks is carried out by the set of instances called Task Manager (TM). The process aids to enhance the overall resource utilization and considerably reducing the processing cost.

The authors [14] stated that different workloads may have different natures lead to the requirement of adaptive scheduling algorithm to accomplish collocated heterogeneous sets of applications with differentiated performance goals. Here, long running jobs are submitted to the system through the job scheduler and then, to the Application Placement Controller (APC). Moreover, they have analyzed the performance model for non interactive workloads and job characteristics.

In addition, load redirect algorithms for cloud systems are enforced in [17]. The ultimate goal of the paper is to reduce the cost of allocated resources regarding virtual machines with Infrastructure-as-a-Service cloud system. It is based on the constraints of Service Level Agreement (SLA), which is noted as threshold on average response time. The distributed solution framed here, integrates the workload prediction mechanism and distributed non-linear optimization mechanisms. Moreover, the authors focused on capacity allocation, load redirection and optimization problems in load balancing on cloud.

In general, cloud computing is a pool of virtualized computer resources that provides dynamically scalable resources. The authors of [18] proposed a soft computing based load balancing approach that affords effective service provisioning. Further, they utilized Stochastic Hill Climbing approach for allocation of incoming jobs to the virtual machines. The steps involved in the approach are given below:

1. Index table of VM servers is maintained with busy state notification
2. When new job arrives in the cloud, query is generated for next allocation
3. Random VM id generation
4. Parsing the allocation table to acquire VM status
    - Allocate the job and update the status in VM table
5. When VM accomplishes processing the task, the cloud receives the result
6. Generation of VM de-allocation notification
7. Continue from step 2 for further allocation

In a different way, ant colony algorithm based load balancing mechanism has been proposed in [19] [22]. The authors of [19] proposed the mechanism related to Mobile Agent Based Open Cloud Computing Federation [MABOCCF], which affords interoperability and portability among different cloud computing platforms. With that concern, the load balancing algorithm has been framed with the combination of ant colony and complex network theory. They concentrated on the process such as under load balancing method, overload load balancing method, update of the pheromone and the evolution of complex network. In [22], the authors proposed Load Balancing Ant Colony Optimization [LBACO] algorithm. The major contribution of this algorithm is to balance the complete system load in order to minimize the make span of the task set. The proposed algorithm inherits the basic ACO algorithm to reduce the computation time of task execution. The steps of the process are given below:

- Initialize Pheromone of Virtual Machine$_i$
- Rule of selecting VM for next task
- Phenomenon Updating

The process also involves in defining the degree of imbalance.

Resource allocation with load balancing criteria has been described by the authors in [20]. The scheduling was based on Join-the Shortest Queue (JSQ) routing rule. They also developed the frame-based non-pre-emptive VM configuration policies, which can be made with optimal throughput by selecting sufficient long frame durations.

The authors of [21] described the load balancing strategy in three-level cloud computing network. The algorithm bonds Opportunistic Load Balancing (OLB) and Load Balancing Min-Min (LBMM) by which it provides better executing efficiency and load balancing. The following figure 4 demonstrates the three-level framework of cloud. The third level is the service node that is responsible for executing the sub task. The second level consists of the service manager that partitions the task into some logical independent sub tasks, whereas the request manager is in the first level, who allocates the task for suitable service manager at the second level.

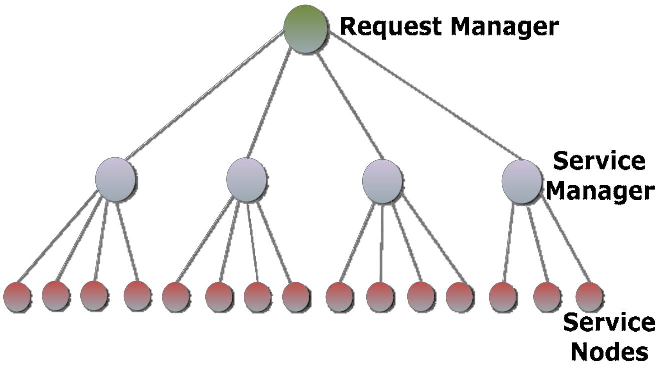

Figure 4: Three-level Cloud Framework

Further, an effective load management system has been proposed in [23] [25]. The load balancing scheme comprises the six following steps.

- Get load status of all nodes
- Evaluate the status of nodes
- Predict the future load flow
- Benefit estimates
- Choose receiver nodes
- Migration

Moreover, the authors also measured the resource utilization to determine the effect of balancing.

Artificial Bee Colony (ABC) algorithm has been proposed for effective load balancing scheme by the authors of [24]. As per the algorithm, the initial process is to calculate the profit of certain requests, which has been set as most needed. Then, it will be updated into the database, which is being considered as equivalent to the dancing area. The remaining requests may explore for information from the database and suitable choice has been made after the comparison. The minimum fitness value of the request is evaluated as follows,

$$Min\ S_{ij} = min\left(w_1 \frac{T_{wait\ i}}{T_{cpu\ i}} + w_2 d_{ij}\right) \qquad (3)$$

Where i=1,2,3,… j=1,2,3,… w_1 and w_2 are the weight ratio, T_{cpu} represents the time taken to run in the CPU for i-th leading bee, T_{wait} denotes waiting time before the i-th leading bee being server, d_{ij} directs the hops for which the i-th leading bee server to the j-th request from the server.

Load balancing in cloud is enforced across different data centres to enhance the network availability and minimize the resource consumption. The paper analyzed the conceit’s efficacy with Hospital Data Management (HDM) in resource consumption and resource availability in cloud computing environment.

The authors of [27] proposed a different approach for efficient load balancing in cloud based on fuzzy logic. It has been claimed that the processor speed and load assignment of virtual machines are used for load balancing in cloud through fuzzy logic. Fuzzy presumption is the process of framing the mapping from given input to expected output using fuzzy logic by which better decisions or patterns can be recognized. The paper results in minimizing the processing time and response time that leads to maximize the resource utilization.

In addition, extension of classic MapReduce model for effective complex job decomposition and task management in cloud has been proposed with the novel view called intelligent load balancing algorithm in [28]. Major contribution of the work was developing an efficient token routing algorithm that serves global state of data distribution in cloud. A heuristic approach for token based load balancing with agent-aid system has also been incorporated with this article. Further, the author [29] developed performance determination criteria of a cloud based load balance in order to solve pare to traffic problem. Figure 5 shows the structural view of load balanced server environment.

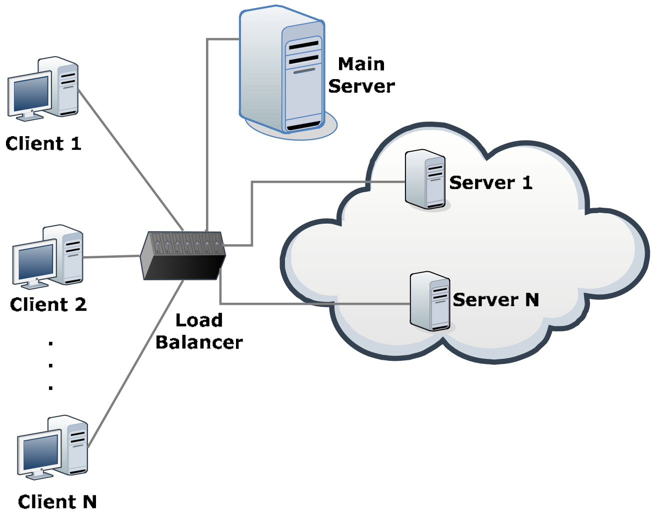

Figure 5: Load Balanced Server Environment

The queuing system is adopted for effectively managing traffic of arrival request to be served by the available servers.

A clear description for load balancing request among cluster of back-end servers is a complicated process [30]. The paper concentrated on three properties namely,

- Randomized partitioning
- Cheap to cache results
- Cost to shift service

It is claimed that cluster may have one front-end node that forwards the queries from client application to the equivalent back-end storage node. The hashing key will be maintained in the back-end node. Caching effect has been analyzed for cluster scaling under different workloads. Moreover, the discussion of the paper was made on front-end caches, network scaling, network latency, imperfect caching policies, non-uniform processing costs and throughput.

C. Metrics For Load Balancing In Cloud

Distinctive parameters examined in the existing load balancing strategies in cloud computing are discussed below.

- *Throughput:* It is used to evaluate the number of tasks executed in a given unit time. Throughput should be high to enhance the system's performance.
- *Fault Tolerance:* It is defined as the ability of an algorithm to accomplish uniform load balancing over the network. An effective load balancing technique should be a good fault tolerant technique.
- *Overhead Associated:* It denotes the amount of overheads enforced during the implantation of load balancing algorithm. Overheads are involved because of the task migration, inter-processor communication and inter-process communication. In general, the presence of overheads should be minimized to improve the performance of load balancing technique.
- *Response Time:* It is defined as the amount of time taken to retort with a load balancing methodology in a distributed cloud environment. The metric should be minimized for effective system's performance.
- *Migration Time:* It is the time taken to move the resources or jobs from one node to another. It should also be reduced for improving the performance of the system.
- *Performance:* It is the parameter used to check the efficiency of the system. The performance of the system has to be enhanced in a considerable manner.
- *Resource Utilization:* Resource utilization should be optimized for efficient load balancing schema.
- *Scalability:* The parameter should be improved. It is defined as the capability of an algorithm to perform balancing loads for a system with any determined number of nodes.

III. RESULTS AND DISCUSSIONS

It is obvious from the previous section that myriad researches developed load balancing technique in distributed networks to resolve the load imbalance problem. Further, the results of this survey are given in the following Table 1 and Table 2.

Table 1: Results of the Survey-Load Balancing Techniques with respect to Networks

Load Balancing in Distributed Networks	Predominant Contribution of Papers
Grid Networks	Deduction of forward and backward ants as a competency ranks and control word identifies the appropriate resource [1] Ant Colony Optimization (ACO) is enforced for optimal configuration [9]
Mesh Network	Load Aware Route Selection (LARS) [2]
Peer-to-Peer Networks	Churn-Resilient Protocol for routine data dissemination [12] Autonomous decentralized resource allocation scheme [3] Self-Acting Load Balancing (SALB) to tackle load imbalance in parallel file system [4] Iterative key redistribution mechanism [11] Novel load balancing algorithm dependent on partial system knowledge [15]
Mobile Ad-hoc	Dynamic Genetic Algorithms to solve dynamic load balancing clustering problem [5]
Wireless Networks	Fuzzy Q-Learning Algorithm [6] Fault Tolerant based Load Balancing Scheme [7]
Overlay Networks	Replica- Aided Load Balancing (RALB) [8]
Cloud Computing	Load Balance Max-Min and Max (LB3M) algorithm [10] Nephele Architecture for parallel processing in cloud [13] Load Redirect Algorithm for cloud system [17] Soft Computing based Load Balancing Approach [12] Combination of Ant Colony and complex Network Theory for Load Balancing [19] Load Balancing Ant Colony Optimization (LBACO) Approach [22] Scheduling based on Join-the Shortest Queue (JSQ) for adaptive Load Balancing [20] Combination of Opportunistic Load Balancing (OLB) and Load Balancing Max-Min (LBMM) [21] Effective Load Management System [23] Artificial Bee Colony based Load Balancing [24] Load Balancing in cloud based on Fuzzy Logic [27] Extension of classic MapReduce model for effective job and task management [28]

Table 2: Results of the Survey with respect to Proposed Architectures in Related Researches

Architecture	Characteristics	Parameters & Metrics	Attained	Not Attained
Based on Tree model and Ant Colony model [1]	Division of jobs based on priorities	Granularity time, tardiness, makespan, processing time, availability, busy rate, current load, power, distribution rate	Competency rank and transitional phases	Enhancing the number of existing resources thereby increasing throughput
Load Aware Route Selection (LARS) [2]	Queuing based	Packet loss, capacity, end-to-end delay	Capacity gain	Proper gateway selection
Decentralized Proactive Resource Allocation (DPRA) [3]	Average Load Level, convex optimization, task scheduling	Throughput, fairness	Adaptability, task migration, resource discovery	Solving practical issues in load migration
Self-Acting Load Balancing (SALB) [4]	Distributed load balancing, dynamic file migration	Mean response time, resource utilization	Scalability, efficiency	Side effects of dynamic file migration
Dynamic Genetic Algorithm (GAs) [5]	Clustering	Immigrants, memory, multi-population	Solves Dynamic Load Balanced Clustering Problem (DLBCP) in MANET	Solving dynamic multi-metric clustering problem in MANET
Fuzzy Q-Learning Algorithm for load balancing [6]	Automatic self-tuning	Call block ratio, network traffic handover	Optimization of Fuzzy Logic Controller (FLC)	Load balancing on $G networks and application on multimedia data services
Fault Tolerant Hybrid Load Balancing Algorithm – AlgHybridLB [7]	Fault tolerant based load balancing	Grid architecture, computer heterogeneity, communication delay, network bandwidth, resource availability, resource unpredictability, job characteristics	Achieves minimum response time, optimal computing node utilization, low complexity	Enhancement in the potentiality of load balancing
Replica-Aided Load Balancing Scheme (RALB) [8]	Replica placement, load balancing, load sharing, performance tuning	Load, message number, link latency	Minimized load balancing, cost efficiency, solves frequency load fluctuations	Application on other distributed network environments
Optimal reconfiguration of distribution system [9]	Ant Colony based Load balancing	Load reduction index, load balancing index, ants behavior. Path selection probability, voltage drop	Real power loss reduction, load balancing	Enhancement of load balancing using distinctive approaches

IV. Conclusion and Future Work

Load balancing is one of the major confrontation in cloud computing. It is required to devote the workload evenly among all the nodes in the network to attain high resource utilization ratio and user satisfaction. Typical data centre implementations rely on large, powerful computing hardware and network infrastructure, which are subject to the usual risks associated with any physical device. With effective load balancing algorithm, the cost can also be reduced in a considerable manner. Existing load balancing techniques that have been analyzed in this paper mainly focused on minimizing service response time and overhead. With that concern, in future enhancement, the need of load balancing in cloud is effectively considered for improving the network performance and scalability with optimized resource utilization.

V. REFERENCES

[1] Leyli Mohammad Khanli, Shiva Razzaghzadeh and Sadegh Vahabzadeh Zargari, “A new step toward load balancing based on competency rank and transitional phases in Grid networks,” In the Journal Proceedings of Future Generation Computer Systems 28, pp.682–688, 2012.

[2] Emilio Ancillotti, Raffaele Bruno, Marco Conti, Antonio Pinizzotto, “Load-aware routing in mesh networks: Models, algorithms and experimentation,” In the Journal Proceedings of Computer Communications 34, pp. 948–961, 2011.

[3] Sheng Di , Cho-Li Wang, “Decentralized proactive resource allocation for maximizing throughput of P2P Grid,” In the Journal Proceedings of Parallel Distrib. Comput. 72, pp. 308–321, 2012.

[4] Bin Dong, Xiuqiao Li, Qimeng Wu, Limin Xiao and Li Ruan, “A dynamic and adaptive load balancing strategy for parallel file system with large-scale I/O servers,” Journal of Parallel Distrib. Comput. 72, pp. 1254–1268, 2012.

[5] Hui Cheng, Shengxiang Yang and Jiannong Cao, “Dynamic genetic algorithms for the dynamic load balanced clustering problem in mobile ad hoc networks,” Journal of Expert Systems with Applications, Vol. 40, Issue. 4, pp. 1381-1392, 2013.

[6] P. Munoz, R. Barco and I. de la Bandera, “Optimization of load balancing using fuzzy Q-Learning for next generation wireless networks,” In the Journal Proceedings of Expert Systems with Applications, Vol. 8, Issue. 11, pp. 1469 – 1479, 2012.

[7] Jasma Balasangameshwara and NedunchezhianRaju, “A hybrid policy for fault tolerant load balancing in grid computing environments,” Journal of Network and Computer Applications, Vol. 35, pp. 412–422, 2012.

[8] Yuehua Wang, ZhongZhou, LingLiu and WeiWu, “Replica-aided load balancing in overlay networks,” Journal of Network and Computer Applications, Vol. 36, Issue. 1, pp. 388–401, 2013.

[9] A.Saffar, R. Hooshmand, A. Khodabakhshian, “A new fuzzy optimal reconfiguration of distribution systems for loss reduction and load balancing using ant colony search-based algorithm,” Journal of Applied Soft Computing, Vol. 11, Issue. 5, pp. 4021–4028, 2011.

[10] Che-Lung, Hung, Hsiao-hsi Wang and Yu-Chen Hu, “Efficient Load Balancing Algorithm for Cloud Computing Network,” 2010

[11] Ioannis Konstantinou, Dimitrios Tsoumakos and Nectarios Koziris, “Fast and Cost-Effective Online Load-Balancing in Distributed Range-Queriable Systems,” IEEE Transactions on Parallel and Distributed Systems, Vol. 22, No. 8, pp. 1350-1364, 2011.

[12] Zhenyu, Li, Gaogang Xie, Kai Hwang and Zhongcheng Li, “Churn-Resilient Protocol for Massive Data Dissemination in P2P Networks,” IEEE Transactions on Parallel and Distributed Systems, Vol. 22, No. 8, pp. 1342- 1349, 2011.

[13] Daniel Warenke and Odej Kao, “Exploiting Dynamic Resource Allocation for Efficient Parallel Data Processing in the Cloud,” IEEE Transactions on Parallel and Distributed Systems, Vol. 22, No. 6, pp. 985- 997, 2011.

[14] David Carrera, Malgorzata Steinder, Ian Whalley, Jordi Torres and Eduard Ayugade, “Autonomic Placement of Mixed Batch and Transactional Workloads,” IEEE Transactions on Parallel and Distributed Systems, Vol. 23, No. 2, pp. 219- 231,2012.

[15] Hung-Chang Hsiao, Hao Liao, Ssu-Ta Chen and Kuo-Chan Huang, “Load Balance with Imperfect Information in Structured Peer-to-Peer Systems,” IEEE Transactions on Parallel and Distributed Systems, Vol. 22, No. 4, 2pp. 634- 649, 2011.

[16] Yunhua Deng and Rynson W.H. Lau, “On Delay Adjustment for Dynamic Load Balancing in Distributed Virtual Environments,” IEEE Transactions on Visualization and Computer Graphics, Vol. 18, No. 4, pp. 529- 537, 2012.

[17] Danilo Ardagna, Sara Casolari, Michele Colajanni and Barbara Panicucci, “Dual time scale distributed capacity allocation and load redirect algorithms for cloud systems,” Journal of Parallel and Distributed Computing, Vol. 72, Issue. 6, pp. 796–808, 2012.

[18] Brototi Mondal, Kousik Dasgupta and Paramartha Dutta, “Load Balancing in Cloud Computing using Stochastic Hill Climbing-A Soft Computing Approach,” 2nd International Conference on Computer, Communication, Control and Information Technology(C3IT-2012), Vol. 4, pp. 783–789,2012.

[19] Zehua Zhang and Xuejie Zhang, “A Load Balancing Mechanism Based on Ant Colony and Complex Network Theory in Open Cloud Computing Federation,” 2nd International Conference on Industrial Mechatronics and Automation, 2010.

[20] Siva Theja Maguluri, R. Srikant and Lei Ying, “Stochastic Models of Load Balancing and Scheduling in Cloud Computing Clusters,” In the Conference Proceedings of INFOCOM, pp. 702-710, 2012.

[21] Shu-Ching Wang, Kuo-Qin Yan, Wen-Pin Liao and Shun-Sheng Wang, “Towards a Load Balancing in a Three-level Cloud Computing Network,” 3rd IEEE International Conference on Computer Science and Information Technology (ICCSIT), Vol. 1, pp. 108-113, 2010.

[22] Kun Li, Gaochao Xu, Guangyu Zhao, Yushuang Dong and Dan Wang, “Cloud Task scheduling based on Load Balancing Ant Colony Optimization,” In the Proceedings of 6th Annual Chinagrid Conference (ChinaGrid), pp 3-4, 2011.

[23] Rui Wang, Wei Le and Xuejie Zhang, “Design and Implementation of an Efficient Load-Balancing Method for Virtual Machine Cluster Based on Cloud Service,” 4th IET International Conference on Wireless, Mobile & Multimedia Networks (ICWMMN 2011), pp. 321-324, 2011.

[24] Jing Yao and Ju-hou He, “Load Balancing Strategy of Cloud Computing based on Artificial Bee Algorithm,” 8th International Conference on Computing Technology and Information Management (ICCM), Vol. 1, 2012, pp. 185-189.

[25] Zenon Chaczko, Venkatesh Mahadevan, Shahrzad Aslanzadeh and Christopher Mcdermid, “Availability and Load Balancing in Cloud Computing,” International Conference on Computer and Software Modeling, Vol. 14, 2011, pp. 134-140.

[26] Pragati Priyadarshinee and Pragya Jain,“Load Balancing and Parallelism in Cloud Computing,” International Journal of Engineering and Advanced Technology, Vol. 1, Issue. 5, 2012, pp. 486-489.

[27] Srinivas Sethi, Anupama Sahu and Suvendu Kumar Jena, “Efficient load Balancing in Cloud Computing using Fuzzy Logic,” IOSR Journal of Engineering (IOSRJEN), Vol. 2, Issue. 7, pp. 65-71, 2012.

[28] Yang Xu, Lei Wu, Liying Guo, Zheng Chen, Lai Yang and Zhongzhi Shi, “An Intelligent Load Balancing Algorithm Towards Efficient Cloud Computing,” In the Proceedings of AAAI Workshop. pp. 27-32,2011.

[29] Ayman G. Fayoumi, “Performance Evaluation Of A Cloud Based Load Balancer Severing Pareto Traffic,” Journal of Theoretical and Applied Information Technology, Vol. 32, No.1, pp. 28- 34, 2011.

[30] Bin Fan, Hyeontaek Lim, David G. Andersen and Michael Kaminsky, “Small Cache, Big Effect: Provable Load Balancing for Randomly Partitioned Cluster Services,” In the Proceedings of 2nd ACM Symposium on Cloud Computing, 2011.

[31] Jinhua Hu, Jianhua Gu, Guofei Sun and Tianhai Zhao, “A Scheduling Strategy on Load Balancing of Virtual Machine Resources in Cloud Computing Environment,” 3rd International Symposium on Parallel Architectures, Algorithms and Programming, pp. 89-96, 2010.

[32] Ali M. Alakeel, “A Guide to Dynamic Load Balancing in Distributed Computer Systems,” IJCSNS International Journal of Computer Science and Network Security, Vol.10 No.6, pp. 153-160, 2010.