International Journal of Technical Research and Applications e-ISSN: 2320-8163,
www.ijtra.com Volume 4, Issue 2 (March-April, 2016), PP. 279-290

SURVEY OF TECHNIQUES AND CHALLENGES FOR LOAD BALANCING IN PUBLIC CLOUD

[1]Karuna G.Bakde, [2]Prof. B .M. Patil
[1]M.E in Computer Network Engineering, [2]Computer Networking Department
Dr. Babasaheb Ambedkar Marathwada University
Ambajogai, Beed.
karunabokade@gmail.com

Abstract— **In present days cloud computing is one of the greatest platform which provides storage of data in very lower cost and available for all time over the internet. But it has more critical issue like security, load management and fault tolerance. In this paper we are discussing Load Balancing approach. Many types of load concern with cloud like memory load, CPU load and network load. Load balancing is the process of distributing load over the different nodes which provides good resource utilization when nodes are overloaded with job. Load balancing has to handle the load when one node is overloaded. When node is overloaded at that time load is distributed over the other ideal nodes. Many algorithms are available for load balancing like Static load balancing and Dynamic load balancing. Load balancing in the cloud computing environment has an important impact on the performance. Good load balancing makes cloud computing more efficient and improves user satisfaction. This article introduces a better load balance model for the public cloud based on the cloud partitioning concept with a switch mechanism to choose different strategies for different situations. In this paper the load balancing strategy to improve the efficiency in the public cloud environment.**

Keywords— load balancing model; public cloud; cloud partition; game theory.

I. INTRODUCTION

Cloud is a technology discontinuity that, with in next 10 years, is likely to dramatically change IT organizational missions, structures, roles, skills and operations. The cloud is changing our life by providing users with new types of services. User gets service from a cloud without paying attention to the details.

Load balancing is the new technique that facilitates networks and resources by a maximum throughput with minimum response time. Proper load balancing can help in utilizing the available resources optimally, thereby minimizing the resource consumption. Load balancing schemes depending on whether the system dynamics are important can be either static or dynamic. Static load balancing scheme divide the traffic equivalently between the services. Dynamic load balancing scheme chooses the lightest server preferred to balance the traffic and selecting an appropriate server needed real time communication with network. The advantage of using dynamic load balancing scheme is that if any node fails, it will not halt the system; it will only affect the system performance. Dynamic load balancer uses policies for keeping track of updated information.

A. CLOUD COMPUTING

Cloud Computing became widespread in last rare year. Cloud provides stretchy software as service (SaaS), Platform as Service(PaaS), Infrastructure as Service (IaaS) [1] . as shown in below fig. .1.

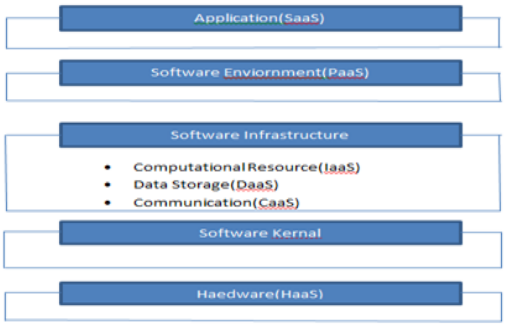

Fig.1. Cloud Computing Architecture.

- **Software as a Service (SaaS):**

SaaS, sometimes referred to as "software on demand". SaaS constitutes a major role in all the cloud computing offerings. Software as a Service (SaaS) is the model in which an application is hosted as a service to customers who access it via the Internet. SaaS is software that is owned, delivered and managed remotely by one or more providers and is offered in a pay-as-per-use manner [2]. SaaS focuses on providing users with business specific capabilities such as e-mail or customer management [3]. The typical user of SaaS offering usually has neither knowledge nor control about the underlying infrastructure [2]. One of the examples of SaaS provider is Google Apps that provides large suite of web based applications for many business applications including accounting, enterprise resource management (ERP), human resource management (HRM), customer relationship management (CRM) and security device manager (SDM).

- **Platform as a Service (PaaS):**

PaaS is a service model cloud computing. In this model, client creates the software using tools and libraries from the

provider[3]. The client controls the applications that run in the environment, but does not control the operating system, hardware and network infrastructure on which they are running [2]. The provider provides the network, servers and storage. One of the examples of PaaS is Google App Engine that provides clients to run their applications on Google's infrastructure [3].PaaS services include application design, development, testing, deployment and hosting. Other services include team collaboration, web service integration, database integration, security, scalability, storage, state management and versioning. PaaS also supports web development interfaces such as Simple Object Access Protocol (SOAP) and Representational State Transfer (REST), which allows the construction of multiple web services, sometimes called smashups. A downfall to PaaS is a lack of interoperability and portability among providers [1].

- **Infrastructure as a Service (IaaS):**

IaaS, also known as cloud infrastructure services, delivers computer infrastructure-typically a platform virtualization outsourced service. IaaS model provides a virtual data center within the cloud. IaaS provides servers (physical and virtualized), cloud-based data storage, etc. The client need not purchase the required servers, data center or the network resources. The key advantage is that customers need to pay only for the time duration they use the service [2]. One of the examples of IaaS providers is Amazon Elastic Compute Cloud (EC2). It provides users with a special virtual machine that can be deployed and run on EC2 infrastructure [3].

The term "cloud" originates from the world of telecommunications when providers began using virtual private network (VPN) services for data communications. The definition of cloud computing provided by National Institute of Standards and Technology (NIST) says that: "Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e. g., networks, servers, storage applications and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction" [4].

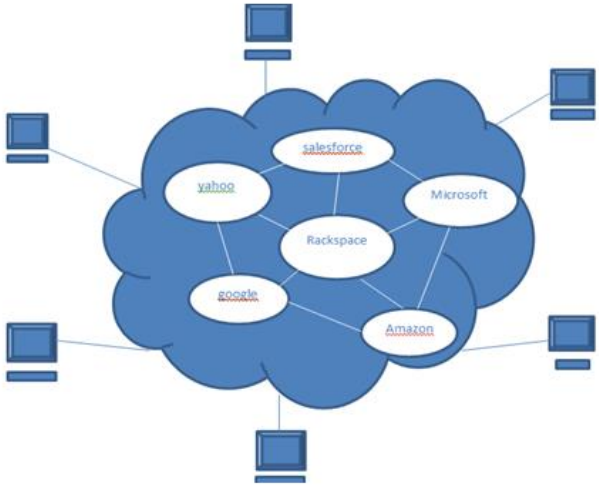

Fig .2. Cloud Computing.

B. Types of Cloud Computing.

PUBLIC

Public clouds are made available to the general public by a service provider who hosts the cloud infrastructure. Generally, public cloud providers like Amazon AWS, Microsoft and Google own and operate the infrastructure and offer access over the Internet. With this model, customers have no visibility or control over where the infrastructure is located. It is important to note that all customers on public clouds share the same infrastructure pool with limited configuration, security protections and availability variances. Public Cloud customers benefit from economies of scale, because infrastructure costs are spread across all users, allowing each individual client to operate on a low-cost, "pay-as-you-go" model. Another advantage of public cloud infrastructures is that they are typically larger in scale than an in-house enterprise cloud, which provides clients with seamless, on- demand scalability. These clouds offer the greatest level of efficiency in shared resources; however, they are also more vulnerable than private clouds [3].

PRIVATE

Private cloud is cloud infrastructure dedicated to a particular organization. Private clouds allow businesses to host applications in the cloud, while addressing concerns regarding data security and control, which is often lacking in a public cloud environment. It is not shared with other organizations, whether managed internally or by a third-party, and it can be hosted internally or externally [3].

HYBRID

Hybrid Clouds are a composition of two or more clouds (private, community or public) that remain unique entities but are bound together offering the advantages of multiple deployment models. In a hybrid cloud, you can leverage third party cloud providers in either a full or partial manner; increasing the flexibility of computing. Augmenting a traditional private cloud with the resources of a public cloud can be used to manage any unexpected surges in workload [5]. Hybrid cloud architecture requires both on-premise resources and off-site server based cloud infrastructure. By spreading things out over a hybrid cloud, you keep each aspect of your business in the most efficient environment possible. The downside is that you have to keep track of multiple cloud security platforms and ensure that all aspects of your business can communicate with each other. Communication with the networks, which will lead to extra traffic added on system. In comparison between these two algorithms, although round robin algorithms based on simple rule, more loads conceived on servers and thus imbalanced traffic discovered as a result. Load balancing schemes depending on whether the system dynamics are important can be either static and dynamic. Static schemes do not use the system information and are less complex while dynamic schemes will bring additional costs for the system but can change as the system status changes. A dynamic scheme is used here for its flexibility [6].

C. LOAD BALANCING

Load Balancing is a technique in which the workload on the resources of a node is shifts to respective resources on the other node in a network without disturbing the running task. A standard way to scale web applications is by using a hardware-based load balancer. The load balancer assumes the IP address

of the web application, so all communication with the web application hits the load balancer first. The load balancer is connected to one or more identical web servers in the back-end. Depending on the user session and the load on each web server, the load balancer forwards packets to different web servers for processing. The hardware-based load balancer is designed to handle high-level of load, so it can easily scale .However, a hardware-based load balancer uses application specific hardware-based components, and thus it is typically expensive. Because of cloud's commodity business model, a hardware-based load balancer is rarely occurred by cloud providers as a service. Instead, one has to use a software based load balancer running on a generic server [9].

D. Types of Load Balancing Algorithms

Static Algorithms:

Static algorithms divide the traffic equivalently between servers. By this approach the traffic on the servers will be disdained easily and consequently it will make the situation more perfectly. This algorithm, which divides the traffic equally, is announced as round robin algorithm. However, there were lots of problems appeared in this algorithm. Therefore, weighted round robin was defined to improve the critical challenges associated with round robin. In this algorithm each servers have been assigned a weight and according to the highest weight they received more connections. In the situation that all the weights are equal, servers will receive balanced traffic [10].

Dynamic Algorithms:

Dynamic algorithms designated proper weights on servers and by searching in whole network a lightest server preferred to balance the traffic. However, selecting an appropriate server needed real time.

E. CHALLENGES IN CLOUD COMPUTING LOAD BALANCING

Before we could review the current load balancing approaches for Cloud Computing, we need to identify the main issues and challenges involved and that could affect how the algorithm would perform. Here we discuss the challenges to be addressed when attempting to propose an optimal solution to the issue of load balancing in Cloud Computing. These challenges are summarized in the following points.

Spatial Distribution of the Cloud Nodes:

Some algorithms are designed to be efficient only for an intranet or closely located nodes where communication delays are negligible. However, it is a challenge to design a load balancing algorithm that can work for spatially distributed nodes. This is because other factors must be taken into account such as the speed of the network links among the nodes, the distance between the client and the task processing nodes, and the distances between the nodes involved in providing the service. There is a need to develop a way to control load balancing mechanism among all the spatial distributed nodes while being able to effectively tolerate high delays [11].

Storage/ Replication:

A full replication algorithm does not take efficient storage utilization into account. This is because the same data will be stored in all replication nodes. Full replication algorithms impose higher costs since more storage is needed. However, partial replication algorithms could save parts of the data sets in each node (with a certain level of overlap) based on each node's capabilities such as processing power and capacity. This could lead to better utilization, yet it increases the complexity of the load balancing algorithms as they attempt to take into account the availability of the data set's parts across the different Cloud nodes [12] .

Algorithm Complexity:

Load balancing algorithms are preferred to be less complex in terms of implementation and operations. The higher implementation complexity would lead to a more complex process which could cause some negative performance issues. Furthermore, when the algorithms require more information and higher communication for monitoring and control, delays would cause more problems and the efficiency will drop. Therefore, load balancing algorithms must be designed in the simplest possible forms [13].

Point of Failure Controlling:

The load balancing and collecting data about the different nodes must be designed in a way that avoids having a single point of failure in the algorithm. Some algorithms (centralized algorithms) can provide efficient and effective mechanisms for solving the load balancing in a certain pattern. However, they have the issue of one controller for the whole system. In such cases, if the controller fails, then the whole system would fail. Any Load balancing algorithm must be designed in order to overcome this challenge . Distributed load balancing algorithms seem to provide a better approach, yet they are much more complex and require more coordination and control to function correctly [14] .

F. SURVEY OF LOAD BALANCING ALGORITHMS

The evolution of cloud can be dated back to the 1960's from the ideas of pioneers like J.C.R Licklider and John

McCarthy. John McCarthy believed that computation cloud organized for public entity. Cloud can be accessed anytime using the internet. So the development of the Internet was one of the major milestone for cloud computing.

In 1969, Leonard Kleinrock , one of the chief scientists of the original Advanced Research Projects Agency Network (ARPANET) which seeded the Internet, said: "As of now, computer networks are still in their infancy, but as they grow up and become sophisticated, I will probably see the spread of "computer utilities" which, like present electric and telephone utilities, will service individual homes and offices across the country." Cloud computing is emerging as a new paradigm of large scale distributed computing. It has moved computing and data away from desktop and portable PCs, into large data centers. It has the capability to harness the power of Internet and wide area network (WAN) to use resources that are available remotely, thereby providing cost effective solution to most of the real life requirements. This connection of networks came to be known as Internet.

Following load balancing techniques are currently prevalent in clouds

- **Round Robin Algorithm**:

Round Robin is a very famous load balancing algorithm, in which the processes are divided between all processors. The process allocation order is maintained locally independent of the allocations from remote processors. In Round Robin, it send the requests to the node with the least number of connections, so at any point of time some node may be heavily loaded and other remain idle [5], this problem is reduced by CLBDM.

- **Central Load Balancing Decision Model (CLBDM):**

CLBDM is a central load balancing decision model, which is suggested by Radojevic and Mario Zagar [9], it's based on session switching at the application layer. The improvement is that, in the cloud it calculated the connection time between the client and the node, and if that connection time exceeds a threshold then connection will be terminated and task will be forwarded to another node using the regular Round Robin rules.

- **MapReduce-based Entity Resolution**:

MapReduce is a computing model and an associated implementation for processing and generating large datasets [10]. Map task and reduce task two main task in this model which written by the user, Map takes an input pair and produces a set of intermediate value pair and Reduce task accepts an intermediate key and a set of values for that key and merges these values to form a smaller set of value. Map task read entities in parallel and process them, this will cause the Reduce task to be overloaded.

- **Ant colony optimization (ACO):**

Kumar Nishant suggested an algorithm [11] of ant colony optimization. In ACO [11] algorithm when the request in initiated the ant start its movement. Movement of ant is of two ways:

Forward Movement : Forward Movement means the ant in continuously moving from one overloaded node to another node and check it is overloaded or under loaded ,if ant find an over loaded node it will continuously moving in the forward direction and check each nodes .Backward Movement: If an ant find an over loaded node the ant will use the back ward movement to get to the previous node, in the algorithm [11] if ant finds the target node then ant will commit suicide, this algorithm reduced the unnecessary back ward movement ,overcome heterogeneity, is excellent in fault tolerance.

- **Load balancing of virtual machine resources:**

J. Hu et al. [12] proposed a scheduling strategy on load balancing of VM resources that uses historical data and current state of the system. This strategy achieves the best load balancing and reduced dynamic migration by using a genetic algorithm. It helps in resolving the issue of load-imbalance and high cost of migration thus achieving better resource utilization.

- **Index Name Server Algorithm (INS):**

The INS algorithm proposed in [13] the goal is to find an algorithm to minimize the data duplication and redundancy. INS is able to handle the load balancing dynamically .INS have some parameters which help in calculating the optimum selection point like that Hash Code of the block of data to be downloaded, the position of the server, the transition quality, the maximum bandwidth. Another calculation point whether the connection can handle additional nodes or not. They classified the busy levels B(a),B(b), and B(c). B(a) means that connection is very busy and cannot handle any additional connection. B(b) means connections is not busy and can handle additional connections. B(c) means that the connection is limited.

- **Opportunistic Load Balancing (OLB):**

Sang proposed OLB is a static load balancing algorithm that has the goal of keeping each node in cloud busy [14]. However OLB does not calculate the execution time of the node, due to this the tasks to be processed in a slower manner and will cause bottlenecks since requests might be pending waiting for nodes to be free.

- **Load Balancing Min-Min Algorithm (LBMM):**

Wang suggested an algorithm called LBMM [15]. LBMM has a three level load balancing framework. In first level LBMM architecture is the request manager which is responsible for receiving the task and assigning it to service manager, when the service manager receives the request; it divides it into subtask and assigns the subtask to a service node based on node availability, remaining memory and the transmission rate which is responsible for execution the task.

- **Dual Direction Downloading Algorithm (DDFTP)**:

DDFTP is a dual direction downloading algorithm from FTP server [16]. This algorithm can be also implemented for Cloud Computing load balancing. This is a fast and efficient concurrent technique for downloading large files from FTP server in a cloud environment. DDFTP uses the concept of processing the files for transfer from two different directions. For example, one server will start from block 0 and keeps downloading incrementally while another server start from block m and keeps downloading in a decrement order. When the two servers download two consecutive blocks, the task is considered as finished and other task can be assigned to the server. As a result, both servers will work independently. The algorithm reduces the network communication between the client and nodes and network overhead.

- **Exponential Smooth Forecast-based on Weighted Lest Connection (ESBWLC):**

The algorithm proposed in [17] is a dynamic load balancing algorithm for cloud computing. ESBWLC build the conclusion of assigning a certain task to a node after having a number of task assigned to that service node and getting to know the node's CPU power, memory, number of connections and the amount of disk space currently in used, then ESBWLC predicts which node is to be selected based on exponential smoothing.

- **A Lock-free multiprocessing solution for LB** :

X. Liu et al. [18] proposed a lock-free multiprocessing load balancing solution that avoids the use of shared memory in contrast to other multiprocessing load balancing solutions which use shared memory and lock to maintain a user session. It is achieved by modifying Linux kernel. This solution helps in improving the overall performance of load balancer in a multi-core environment by running multiple load-balancing processes in one load balancer.

- **Honeybee Foraging Behavior** :

M. Randles et al. [19] investigated a decentralized honeybee-based load balancing technique that is a nature-inspired algorithm for self-organization. It achieves global load balancing through local server actions. Performance of the system is enhanced with increased system diversity but throughput is not increased with an increase in system size. It is best suited for the conditions where the diverse population of service types is required.

- **VectorDot:**

A. Singh et al. [16] proposed a novel load balancing algorithm called VectorDot. It handles the hierarchical complexity of the data-center and multidimensionality of resource loads across servers, network switches, and storage in an agile data center that has integrated server and storage virtualization technologies. VectorDot uses dot product to distinguish nodes based on the item requirements and helps in removing overloads on servers, switches and storage nodes.

- **CARTON:**

R. Stanojevic et al. [17] proposed a mechanism CARTON for cloud control that unifies the use of LB and DRL. LB (Load Balancing) is used to equally distribute the jobs to different servers so that the associated costs can be minimized and DRL (Distributed Rate Limiting) is used to make sure that the resources are distributed in a way to keep a fair resource allocation. DRL also adapts to server capacities for the dynamic workloads so that performance levels at all servers are equal. With very low computation and communication overhead, this algorithm is simple and easy to implement.

- **Compare and Balance:**

Y. Zhao et al. [18] addressed the problem of intra-cloud load balancing amongst physical hosts by adaptive live migration of virtual machines. A load balancing model is designed and implemented to reduce virtual ma-chines' migration time by shared storage, to balance load amongst servers according to their processor or IO usage, etc. and to keep virtual machines' zero-downtime in the process. A distributed load balancing algorithm COMPARE AND BALANCE is also proposed that is based on sampling and reaches equilibrium very fast. This algorithm assures that the migration of VMs is always from high-cost physical hosts to low-cost host but assumes that each physical host has enough memory which is a weak assumption.

- **Event-driven:**

V. Nae et al. [19] presented an event-driven load balancing algorithm for real-time Massively Multiplayer Online Games (MMOG). This algorithm after receiving capacity events as input, analyzes its components in context of the resources and the global state of the game session, thereby generating the game session load balancing actions. It is capable of scaling up and down a game session on multiple resources according to the variable user load but has occasional QoS breaches.

- **Scheduling strategy on LB of VM resources:**

J. Hu et al. [20] proposed a scheduling strategy on load balancing of VM resources that uses historical data and current state of the system. This strategy achieves the best load balancing and reduced dynamic migration by using a genetic algorithm. It helps in resolving the issue of load imbalance and high cost of migration thus achieving better resource utilization.

- **CLBVM:**

A. Bhadani et al. [21] proposed a Central Load Balancing Policy for Virtual Machines (CLBVM) that balances the load evenly in a distributed virtual machine/cloud computing environment. This policy improves the overall performance of the system but does not consider the systems that are fault-tolerant.

- **LBVS:**

H. Liu et al. [22] proposed a load balancing virtual stor-age strategy (LBVS) that provides a large scale net data stor-age model and Storage as a Service model based on Cloud Storage. Storage virtualization is achieved using an architecture that is three-layered and load balancing is achieved using two load balancing modules. It helps in improving the efficiency of concurrent access by using replica balancing further reducing the response time and enhancing the capacity of disaster recovery. This strategy also helps in improving the use rate of storage resource, flexibility and robustness of the system.

- **Task Scheduling based on LB:**

Y. Fang et al. [23] discussed a two-level task scheduling mechanism based on load balancing to meet dynamic requirements of users and obtain high resource utilization. It achieves load balancing by first map-ping tasks to virtual machines and then virtual machines to host resources thereby improving the task response time, resource utilization and overall performance of the cloud computing environment.

- **Honeybee Foraging Behavior:**

M. Randles et al. [24] investigated a decentralized honeybee-based load balancing technique that is a nature-inspired algorithm for self-organization. It achieves global load balancing through local server actions. Performance of the system is enhanced with increased sys-tem diversity but throughput is not increased with an increase in system size. It is best suited for the conditions where the diverse population of service types is required.

- **Biased Random Sampling:**

M. Randles et al. [24] investigated a distributed and scalable load balancing approach that uses random sampling of the system domain to achieve self-organization thus balancing the load across all nodes of the system. The performance of the system is improved with high and similar population of resources thus resulting in an in-creased throughput by effectively utilizing the increased sys-tem resources. It is degraded with an increase in population diversity.

- **Active Clustering:**

M. Randles et al. [24] investigated a self-aggregation load balancing technique that is a self-aggregation algorithm to optimize job assignments by connecting similar services using local re-wiring. The performance of the system is enhanced with high resources thereby in-creasing the throughput by using these resources effectively. It is degraded with an increase in system diversity.

- **ACCLB:**

Z. Zhang et al. [25] proposed a load balancing mechanism based on ant colony and complex network theory (ACCLB) in an open cloud computing federation. It uses small-world and

scale-free characteristics of a complex network to achieve better load balancing. This technique overcomes heterogeneity, is adaptive to dynamic environments, is excel-lent in fault tolerance and has good scalability hence helps in improving the performance of the system.

- **(OLB + LBMM):**

S.-C. Wang et al. [26] proposed a two-phase scheduling algorithm that combines OLB (Opportunistic Load Balancing) and LBMM (Load Balance Min-Min) scheduling algorithms to utilize better executing efficiency and maintain the load balancing of the system. OLB scheduling algorithm, keeps every node in working state to achieve the goal of load balance and LBMM scheduling algorithm is utilized to minimize the execution time of each task on the node thereby minimizing the overall completion time. This combined approach hence helps in an efficient utilization of resources and enhances the work efficiency.

- **Decentralized content aware:**

H. Mehta et al. [27] proposed a new content aware load balancing policy named as work-load and client aware policy (WCAP). It uses a parameter named as USP to specify the unique and special property of the requests as well as computing nodes. USP helps the scheduler to decide the best suitable node for processing the requests. This strategy is implemented in a decentralized manner with low overhead. By using the content information to narrow down the search, it improves the searching performance overall performance of the system. It also helps in reducing the idle time of the computing nodes hence improving their utilization.

- **Server-based LB for Internet distributed services:**

A. M. Nakai et al. [28] proposed a new server-based load balancing policy for web servers which are distributed all over the world. It helps in reducing the service response times by using a protocol that limits the redirection of requests to the closest remote servers without overloading them. A middleware is described to implement this protocol. It also uses a heuristic to help web servers to endure overloads.

- **Join-Idle-Queue:**

Y. Lua et al. [29] proposed a Join-Idle-Queue load balancing algorithm for dynamically scalable web services. This algorithm provides large-scale load balancing with distributed dispatchers by, first load balancing idle processors across dispatchers for the availability of idle processors at each dispatcher and then, assigning jobs to processors to reduce average queue length at each processor. By removing the load balancing work from the critical path of request processing, it effectively reduces the system load, incurs no communication overhead at job arrivals and does not increase actual response time.

- **Lock-free multiprocessing solution for LB:**

X. Liu et al. [30] proposed a lock-free multiprocessing load balancing solution that avoids the use of shared memory in contrast to other multiprocessing load balancing solutions which use shared memory and lock to maintain a user session. It is achieved by modifying Linux kernel. This solution helps in improving the overall performance of load balancer in a multi-core environment by running multiple load-balancing processes in one load balancer.

- **Green cloud computing :**

Xiong, N., et al.[31], are especially cognizant that green cloud computing (GCC) is an expansive extent and a sizzling field. The distinction between "consumer and provider", of cloud-based energy resources is important in creating a world-wide ecosystem of GCC. A client essentially submits its service request to the cloud service provider with the association of Internet or wired/wireless systems. The result of the requested service is conveyed once more to the client in time, though the qualified information space and process, interoperating methodologies, administration organization, correspondences and appropriated registering, are all easily intuitive by the systems

- **Weight less connection algorithm:**

Ren. [31], proposed algorithm is called ESWLC (Exponential Smooth Forecast based on Weighted Least Connection). ESWLC improves weight less connection by taking into account the time series and trials. That is ESWLC builds the decision based on the experience of the node's CPU power, memory, number of connections and the amount of disk space currently being used. ESWLC then predicts which node is to be selected based on exponential smoothing**.**

- **Reverse Filling Load Balance algorithm**:

D.choudhary and R.singh[33], proposed Reverse Filling Load Balance algorithm, it reduces the waiting time of the machines along with the overheads in the system resulting in lesser cost than existing approach.

- **Effective load balancing algorithm:**

P.Amaravathi and G.Devi Himaja[34], proposed an effective load balancing algorithm using Ant colony optimization technique to maximize or minimize different performance parameters like CPU load, Memory capacity, Delay or network load for the clouds of different sizes.

- **Token generation algorithm**:

S.D.Sonwane and R.H.Borhade[35],proposed Token Generation Algorithm for file uploading. The system will get the load of all cloud systems to perform functions as the client while Controller will get the load of all balancers and send it to the user by establishing connection to balancers. After this, by making connection to servers, balancer gets the load of all servers and sends it to the Controller. Client will upload the files on the cloud by using load balancing algorithm to controller who uploads the file to the balancer who has a minimum load. Next, balancer uploads the file to the server that has a minimum load. Client can Search and Download the file and Deployments from the controller who will search and download the file from balancers.

II. SYSTEM MODEL

There are several cloud computing categories with this work focused on a public cloud. A public cloud is based on the standard cloud computing model, with service provided by a service provider. A large public cloud will include many nodes and the nodes in different geographical locations. Cloud partitioning is used to manage this large cloud. A cloud partition is a subarea of the public cloud with divisions based on the geographic locations. The architecture is shown in Fig.5.

The load balancing strategy is based on the cloud partitioning concept .After creating the cloud partitions, the load balancing then starts: when a job arrives at the system, with the main controller deciding which cloud partition should receive the job. The partition load balancer then decides how to assign the jobs to the nodes. When the load status of a cloud partition is normal, this partitioning can be accomplished locally. If the cloud partition load status is not normal, this job should be transferred to another partition.

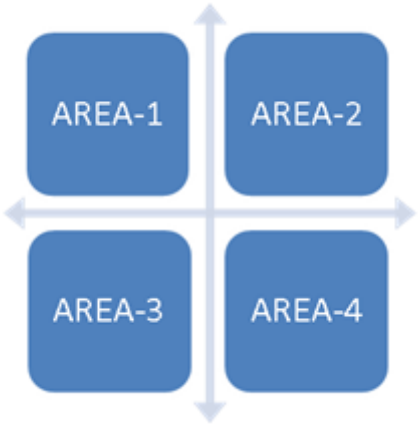

Fig.5 Simple View of Cloud Partitioning.

The load balancing strategy is based on the cloud partitioning concept. After creating the cloud partitions, the load balancing it starts: when a job arrives at the system, with the main controller deciding which cloud partition should receive the job. The partition load balancer then decides how to assign the jobs to the nodes. When the load status of a cloud partition is normal, this partitioning can be accomplished locally. If the cloud partition load status is not normal, this job should be transferred to another partition [36].

A. existing methodology

A public cloud is one based on the standard cloud computing model, in which a service provider makes resources, such as applications and storage, available to the general public over the Internet. Public cloud is made up of several nodes situated in deferent geographic location. Cloud partitioning is a method to make partitions of huge public cloud is some segment of cloud. A cloud partition has several nodes belongs to a particular area, these subarea of the public cloud based on the geographic locations. These subareas are considered to be as cluster of nodes with a load manages. Model of the cloud partitioning is depicted on the following figure. In this model load balancing is implemented in two steps, in first steps public cloud is partitioned into four subarea named Partition#1, Partition#2, Partition#3 and Partition#4. Each partition has a Load balancer (LB) associated with multiple multi-core nodes. There is a main controller system which manages the load balancer called Load Balancer Manager (LBM).

Cloud partition state gather all status information from the node and compute load degree of the partition. Load degree is depend on various static and dynamic parameters. Static parameter is memory speed of CPU, number of CPU include in load balancing and memory size. Dynamic parameters are CPU utilization ratio, Network bandwidth, memory utilization ratio.

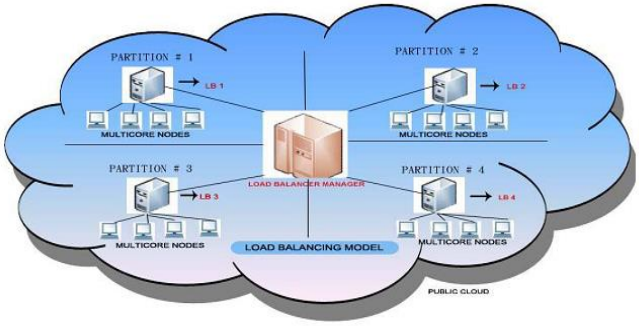

Fig.6 Cloud partitioning model with nodes.

There are several cloud computing categories with this work focused on a public cloud. A public cloud is based on the standard cloud computing model, with service provided by a service provider. A large public cloud will include many nodes and the nodes in different geographical locations. Cloud partitioning is used to manage this large cloud. A cloud partition is a subarea of the public cloud with divisions based on the geographic locations [37].

B. Cloud Partition Load Balancing Strategy

Good load balance will improve the performance of the entire cloud. However, there is no common method that can adapt to all possible different situations. Various methods have been developed in improving existing solutions to resolve new problems. Each particular method has advantage in a particular area but not in all situations. Therefore, the current model integrates several methods and switches between the load balance method based on the system status. A relatively simple method can be used for the partition idle state with a more complex method for the normal state. The load balancers then switch methods as the status changes. Here, the idle status uses an improved Round Robin algorithm while the normal status uses a game theory based load balancing strategy.

C. Load balance strategy for the idle status

When the cloud partition is idle, many computing resources are available and relatively few jobs are arriving. In this situation, this cloud partition has the ability to process jobs as quickly as possible so a simple load balancing method can be used. There are many simple load balance algorithm methods such as the Random algorithm, the Weight Round Robin, and the Dynamic Round Robin [38]. The Round Robin algorithm is used here for its simplicity.

The Round Robin algorithm is one of the simplest load balancing algorithms, which passes each new request to the next server in the queue. The algorithm does not record the status of each connection so it has no status information. In the regular Round Robin algorithm, every node has an equal opportunity to be chosen. However, in a public cloud, the

configuration and the performance of each node will be not the same; thus, this method may overload some nodes. Thus, an improved Round Robin algorithm is used , which called "Round Robin based on the load degree evaluation".

The algorithm is still fairly simple. Before the Round Robin step, the nodes in the load balancing table are ordered based on the load degree from the lowest to the highest. The system builds a circular queue and walks through the queue again and again. Jobs will then be assigned to nodes with low load degrees. The node order will be changed when the balancer refreshes the Load Status Table.

However, there may be read and write inconsistency at the refresh period *T*. When the balance table is refreshed, at this moment, if a job arrives at the cloud partition, it will bring the inconsistent problem. The system status will have changed but the information will still be old. This may lead to an erroneous load strategy choice and an erroneous nodes order. To resolve this problem, two Load Status Tables should be created as: Load Status Table 1 and Load Status Table 2. A flag is also assigned to each table to indicate Read or Write. When the flag = "Read", then the Round Robin based on the load degree evaluation algorithm is using this table. When the flag = "Write", the table is being refreshed, new information is written into this table. Thus, at each moment, one table gives the correct node locations in the queue for the improved Round Robin algorithm, while the other is being prepared with the updated information. Once the data is refreshed, the table flag is changed to "Read" and the other table's flag is changed to "Write". The two tables then alternate to solve the inconsistency.

D. Load balancing strategy for the normal status

When the cloud partition is normal, jobs are arriving much faster than in the idle state and the situation is far more complex, so a different strategy is used for the load balancing. Each user wants his jobs completed in the shortest time, so the public cloud needs a method that can complete the jobs of all users with reasonable response time.

Penmatsa and Chronopoulos [39] proposed a static load balancing strategy based on game theory for distributed systems. And this work provides us with a new review of the load balance problem in the cloud environment. As an implementation of distributed system, the load balancing in the cloud computing environment can be viewed as a game.Game theory has non-cooperative games and cooperative games. In cooperative games, the decision makers eventually come to an agreement which is called a binding agreement. Each decision maker decides by comparing notes with each others. In non-cooperative games, each decision maker makes decisions only for his own benefit. The system then reachs the Nash equilibrium, where each decision maker makes the optimized decision. The Nash equilibrium is when each player in the game has chosen a strategy and no player can benefit by changing his or her strategy while the other players strategies remain unchanged. Game theory has non-cooperative games and cooperative games. In cooperative games, the decision makers eventually come to an agreement which is called a binding agreement. Each decision maker decides by comparing notes with each other's. In non-cooperative games, each decision maker makes decisions only for his own benefit.

There have been many studies in using game theory for the load balancing. Grosu et al[40] . proposed a load balancing strategy based on game theory for the distributed systems as a non-cooperative game using the distributed structure. They compared this algorithm with other traditional methods to show that their algorithm was less complexity with better performance. Aote and Kharat[41] gave a dynamic load balancing model based on game theory. This model is related on the dynamic load status of the system with the users being the decision makers in a non-cooperative game.

III. PROPOSED METHODOLOGY

Improved Load balancing algorithm is used here. In this algorithm the client first requests the load balancer to find a suitable Virtual Machine to perform the required operation. It is shown in fig.7. The process first starts by maintain a list of all the VMs each row is individually indexed to speed up the lookup process.

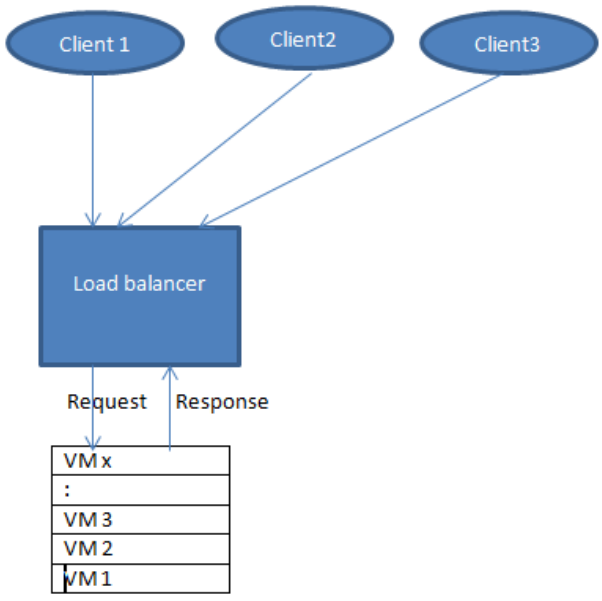

Fig 7: load balancing algorithm

If a match is found on the basis of size and availability of the machine, then the load balancer accepts the request of the client and allocates that VM to the client. If, however there is no VM available that matches the criteria then the load balancer returns -1 and the request is queued.

IV. METHODOLOGY

A. Main controller and balancers

The load balance solution is done by the main controller and the balancers. The main controller first assigns jobs to the suitable cloud partition and then communicates with the balancers in each partition to refresh this status information. Since the main controller deals with information for each partition, smaller data sets will lead to the higher processing rates. The balancers in each partition gather the status information from every node and then choose the right strategy to distribute the jobs. The relationship between the balancers and the main controller is shown in Fig 8.

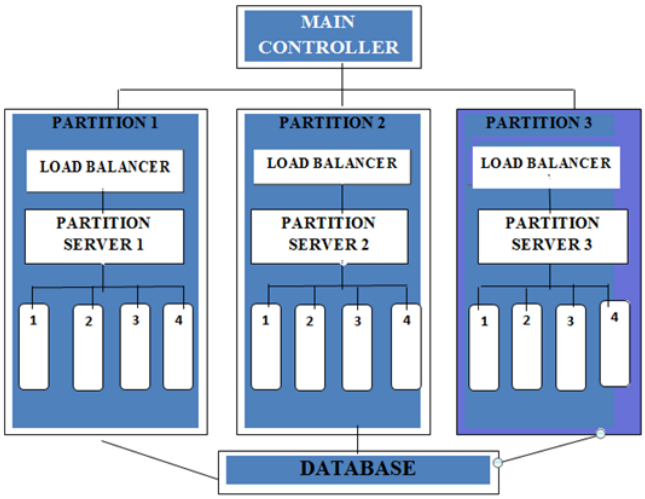

Fig 8: Relationship between balancer and main controller.

6.1 Assigning jobs to the cloud partition

The cloud partition status can be divided into three types:

(1)**Idle**: When the percentage of idle nodes exceeds α, change to idle status.

(2) **Normal**: When the percentage of the normal nodes exceeds β, change to normal load status.

(3) **Overload**: : When the percentage of the overloaded nodes exceedsγ, change to overloaded status. The parameters α, β, and γ are set by the cloud partition balancers.

When a job arrives at the public cloud, the first step is deciding to choose right partition.

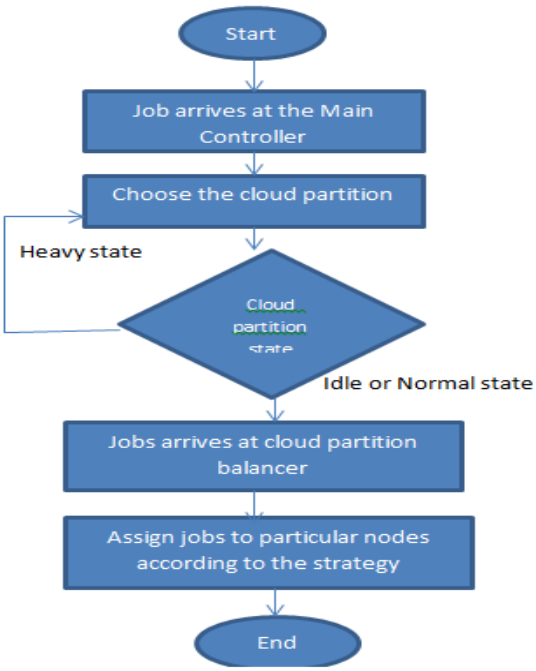

Fig.9: Job Assignment Strategy

6.2 ASSIGNING JOB TO THE SUITABLE CLOUD PARTITION

Cloud partition state gather all status information from the node and compute load degree of the partition. Load degree is depend on various static and dynamic parameters. Static parameter is memory speed of CPU, number of CPU include in load balancing and memory size. Dynamic parameters like CPU utilization ratio, Network bandwidth, memory utilization ratio , the CPU utilization ratio, the network bandwidth, etc

BEST PARTITION SEARCHING ALGORITHM

```
begin
 while job do
   searchBestPartition (job);
     if partitionState == idle || partitionState == normal then
      Send Job to Partition;
    else
      search for another Partition;
    end if
    end while
    end
```

A cloud partition has several nodes belongs to a particular area. Each partition has a Load balancer (LB) associated with multiple multi-core nodes. There is a main controller system which manages the load balancer called Load Balancer Manager (LBM). After partitioning the public cloud into different partitions, load balancing then starts.

6.2.1 WORKING OF LOAD BALANCER MANAGER (LBM):

In this model, Load Balancer Manager (LBM) is responsible for the following task-

- Receives the jobs from different end users.
- Choose a specific partition for the received jobs.
- Check the status of the cloud partition (Status may be in one of these: IDLE, NORMAL, and HEAVY).
- If the partition Status=HEAVY the no allocation will be done, it means all nodes are overloaded already.

If partition Status=IDLE or NORMAL then forward the Jobs to the respective Load balancer (LB). Now the load balancer activated and starts its work.

6.2.2 POSSIBLE STATUS OF CLOUD PARTITION:

Cloud partition can be in one of the following three statuses –

- IDLE – In this, most of the nodes are in idle state.
- NORMAL – In this status, some of the nodes are in idle status while some others are overloaded.
- HEAVY – In this status of the cloud partition, most of the nodes are overloaded.

Load degree is computed from these parameters are as follows:

Compute Load Degree

Inputs:

The static parameters include the number of CPU's, the CPU processing speeds, the memory size, etc. Dynamic parameters are the memory utilization ratio, the CPU utilization ratio, the network bandwidth.

Process:

1. Define a load parameter set: F = {F1; F2… Fm } with each Fi ($1 \leqslant i \leqslant m$,Fi □[0,1]) parameter being either static or dynamic. m represents the total number of the parameters.

2. Compute the load degree as:

Load _degree (N) = Fi

=(i =1) are weights that may differ for different kinds of jobs. N represents the current node.

3. Average cloud partition degree from the node load degree statistics as:

Load _degree=avg=

4. Three level node status are defined

IDLE:

Load_degree(N)=0

NORMAL:

$0<$Load_degree(N) $\leqslant$ Load_degree(N) high .

OVERLOAD:

Load_degree(N) high .$\leqslant$ Load_degree(N)

Output :-

Idle or Normal Or Overloaded.

6.3 CLOUD PARTITION AND LOAD BALANCING STRATEGY

Good load balancing algorithm improving the performance of cloud. Cloud partition state gather all status information from the node and compute load degree of the every partition. When a new job arrives, according the status of load degree which partition is suitable is decided. Balancer for each cloud partition chooses the best load balancing strategy.

Load balancing algorithm gives the better performance then Round Robin algorithm. This algorithm implements a th load balancer (LB) to monitor the loads on each VM. Here each VM is assigned to only one task at a time and can be assigned another task only when the current task has completed successfully. Load balancing algorithm is completely based on virtual machine. In this algorithm client first requesting the load balancer to check the right virtual machine which access that load easily and perform the operations which is given by the client or user. The process first starts by maintaining a list of virtual machines each row is individually indexed to speed up the look up process. The job of TLB is to maintain an index table of all VMs as well as their current states (Available or Busy). If match is found on the basis of size and availability of the machine, then the load balancer accepts the request of the client and allocates that virtual machine to the client [4]. The TLB scans the index table from top to bottom until the first available VM is found. If it finds, then TLB returns the VM id to the Data Center Controller(DCC). The Data Centre communicates the request to the VM identified by the id. On the other hand, if the TLB doesn't find any VM in the available state it simply returns null. In this case DCC queues the request until the availability of any VM.

This load balancing algorithm is improved load balancing for public cloud. So, this algorithm is named as Improved throttled load balancing algorithm for public cloud.

Improved throttled load balancing algorithm steps using in public cloud are as follows:

6.4 IMPROVED THROTTLED LOAD BALANCING ALGORITHM FOR PUBLIC CLOUD

Step1: Main Controller maintains the status information of all partitions

Step2: Vm Load Balancer maintains an index table of VMs and the state of the VM (BUSY/AVAILABLE). At the start all partition and its VM's are available.

Where, Vm=Virtual Machine.

Step3: While (Vm_clicks=0) do

Step4: Evaluate fitness of Load efficiency.

Where, n = total no. of users

i = no. of Vm

Step5: Searching for new Vm

Step6: Select the fittest Vm and Calculate Load.

Step7: Calculate Load ,check if the load on selected Vm is idle, normal or overloaded.

Step8: D0 Vm allocation or task shifting in continuous loop.

V. CONCLUSION:-

This paper demonstrated the applicability of using partition techniques and Improved throttled Load Balancing Algorithm to obtain measurable improvements in resource utilization and availability of cloud-computing environment and increase the business performance in cloud based sector.

VI. FUTURE WORK

In this paper Improved throttled load balancing algorithm are used. Other load balancing strategy give better result and improve the performance. A good algorithm is needed for setting the load degree high and low.

REFERENCES

1. Klaithem Al Nuaimi, Nader Mohamed, Mariam Al Nuaimi and Jameela Al-Jaroodi" A Survey of Load Balancing in Cloud Computing: Challenges and Algorithms" 2012 IEEE Second Symposium on Network Cloud Computing and Applications.
2. J. Srinivas, K.V.S.Reddy and A.M. Qyser, "Cloud Computing Basics", International Journal of Advanced Research in Computer and Communication Engineering, 1(5), July 2012.
3. S. Ray and A.D. Sarkar, "Execution Analysis of Load Balancing Algorithms in Cloud Computing Environment", International Journal on Cloud Computing Services and Architecture, 2(5), October 2012.
4. P. Mell and T. Grance, "The NIST definition of cloud computing", http://csrc.nist.gov/ publications/nistpubs/800- 145/SP800-145.pdf, 2012.
5. http://www..e2networks.com/cloud-servers-india/cloud-computing-architecture
6. http://www.techopedia.com/definition/26559/community-cloud
7. Martin Randles, David Lamb and A. Taleb-Bendiab, "A Comparative Study into Distributed Load Balancing Algorithms for Cloud Computing", 2010 IEEE 24th International Conference on Advanced Information Networking and Applications Workshops.
8. Ram Prasad Padhy and P Goutam Prasad Rao ,"Load Balancing In Cloud Computing system", Rourkela-769 008, Orissa, India May, 2011.
9. K. Nishant, P. Sharma, V. Krishna, C. Gupta, K. P. Singh,N. Nitin, and R. Rastogi, "Load balancing of nodesin cloud using ant colony optimization", in Proc. 14thInternational Conference on Computer Modelling andSimulation (UKSim), Cambridgeshire, United Kingdom,Mar. 2012, pp. 28-30
10. Ms. Parin V. Patel, Mr. Hitesh. D. Patel and Pinal. J. Patel, "A Survey On Load Balancing In Cloud Computing", International Journal of Engineering Research & Technology (IJERT)Vol. 1 Issue 9, November- 2012ISSN: 2278-0181
11. Buyya R., R. Ranjan and RN. Calheiros, "InterCloud: Utility-oriented federation of cloud computing environments for scaling of application services," in proc. 10th International Conference on Algorithms and Architectures for Parallel Processing (ICA3PP), Busan, South Korea, 2010.
12. Foster, I., Y. Zhao, I. Raicu and S. Lu, "Cloud Computing and Grid Computing 360-degree compared," in proc. Grid Computing Environments Workshop, pp: 99-106, 2008.
13. Grosu, D., A.T. Chronopoulos and M. Leung, "Cooperative load balancing in distributed systems," in Concurrency and Computation: Practice and Experience", Vol. 20, No. 16, pp: 1953-1976, 2008.
14. Ranjan, R., L. Zhao, X. Wu, A. Liu, A. Quiroz and M. Parashar, "Peer to- peer cloud provisioning: Service discovery and load-balancing," in Cloud Computing - Principles, Systems and Applications, pp: 195-217,2010.
15. L. Kleinrock, "A Vision For The Internet St Journal Of Research", Nov, 2005,(1),Pg4-5.
16. Singh A., Korupolu M. and Mohapatra D. (2008) ACM/IEEE conference on Supercomputing.
17. Stanojevic R. and Shorten R. (2009) IEEE ICC, 1-6.
18. Zhao Y. and Huang W. (2009) 5th International Joint Confer-ence on INC, IMS and IDC, 170-175.
19. Nae V., Prodan R. and Fahringer T. (2010) 11th IEEE/ACM International Conference on Grid Computing (Grid), 9-17.
20. Hu J., Gu J., Sun G. and Zhao T. (2010) 3rd International Symposium on Parallel Architectures, Algorithms and Pro-gramming, 89-96.
21. Bhadani A. and Chaudhary S. (2010) 3rd Annual ACM Banga-lore Conference.
22. Liu H., Liu S., Meng X., Yang C. and Zhang Y. (2010) Interna-tional Conference on Service Sciences (ICSS), 257-262.
23. Fang Y., Wang F. and Ge J. (2010) Lecture Notes in Comput-er Science, 6318, 271-277.
24. Randles M., Lamb D. and Taleb-Bendiab A. (2010) 24th Inter-national Conference on Advanced Information Networking and Applications Workshops, 551-556.
25. Zhang Z. and Zhang X. (2010) 2nd International Conference on Industrial Mechatronics and Automation, 240-243.
26. Wang S., Yan K., Liao W. and Wang S. (2010) 3rd Internation-al Conference on Computer Science and Information Technol-ogy, 108-113.
27. Mehta H., Kanungo P. and Chandwani M. (2011) International Conference Workshop on Emerging Trends in Technology, 370-375.
28. Nakai A.M., Madeira E. and Buzato L.E. (2011) 5th Latin-American Symposium on Dependable Computing, 156-165.
29. Lua Y., Xiea Q., Kliotb G., Gellerb A., Larusb J. R. and Green-ber A. (2011) Int. Journal on Performance evaluation.
30. Liu Xi., Pan Lei., Wang Chong-Jun. and Xie Jun-Yuan. (2011) 3rd International Workshop on Intelligent Systems and Appli-cations, 1-4.
31. Xiong, N.; Han, W.; Vandenberg, A., "Green cloud computing schemes based on networks: a survey," Communications, IET , vol.6, no.18, pp.3294,3300, Dec. 18 2012

32. C. Wang, Q. Wang, K. Ren, and W. Lou, "Privacy-preserving public auditing for storage security in cloud computing," in Proc. of IEEE INFOCOM'10, March 2010.
33. D.choudhary and R.singh," A New Load Balancing Technique for Virtual Machine Cloud Computing Environment" in International Journal of Computer Applications (0975 – 8887) Volume 69– No.23, May 2013
34. P.Amaravathi and G.Devi Himaja," Effective Load Balancing Technique In Pcloud"in International Journal of Research in Computer and Communication Technology, Vol 3, Issue 11, November – 2014
35. Snehal D. SonawaneÀ and R. H.Borhade," Load Distribution and Balancing over Cloud using Cloud Partitioning "in International Journal of Current Engineering and Technology E-ISSN 2277 – 4106, P-ISSN 2347 – 5161-2014 INPRESSCO
36. Gaochao Xu, Junjie Pang, and Xiaodong Fu, A Load Balancing Model Based on Cloud Partitioning for the Public Cloud, IEEE TRANSACTIONS ON CLOUD COMPUTING YEAR 2013.
37. A. Rouse, Public cloud, http://searchcloudcomputing.techtarget.com/definition/public-cloud, 2012.
38. D. MacVittie, Intro to load balancing for developers —The algorithms, https://devcentral.f5.com/blogs/us/introto-load-balancing-for-developers-ndash-the-algorithms,2012.
39. S. Penmatsa and A. T. Chronopoulos, Game-theoretic static load balancing for distributed systems, Journal of Parallel and Distributed Computing, vol. 71, no. 4,pp. 537-555, Apr. 2011.
40. D. Grosu, A. T. Chronopoulos, and M. Y. Leung, Load balancing in distributed systems: An approach using cooperative games, in Proc. 16th IEEE Intl. Parallel and Distributed Processing Symp., Florida, USA, Apr. 2002,pp. 52-61.
41. S. Aote and M. U. Kharat, A game-theoretic model for dynamic load balancing in distributed systems, in Proc. The International Conference on Advances in Computing, Communication and Control (ICAC3 '09), New York, USA, 2009, pp. 235-238.