IJCSC 0973-7391 Vol- 6 • Issue-1 Sep - Mar 2015 pp.112-120 **Impact Factor-2.5** available at www.csjournalss.com

Perspective Study on Load Balancing Paradigms in Cloud Computing

Jitendra B Bhatia, Malaram K Kumhar, Dr. Madhuri Bhavsar
Department of Computer Science, Nirma University, Ahmedabad, India
{jitendra.bhatia,malaram.kumhar, madhuri.bhavsar}@nirmauni.ac.in

Abstract: Due to the growth of IT industry, the need of computing and storage has increased manifolds. Cloud Computing is the emerging technology for online allotment of computing resources and storage for user's data on the pay-as-you-go basis following utility computing model. Cloud computing is a general term for the delivery of hosted services over the Internet and includes virtualization, networking, utility computing, distributed computing, software and web services. Considering the growing importance of cloud, it must provide high performance gain to the user and must be beneficial for the Cloud Service Provider at the same time. With this goal, there are challenges ahead. Load balancing being one of them which helps meeting the QoSbenchmark that the user requires and on the other hand maximizes the provider's profit by optimum use of resources.In order to balance the load in cloud the resources and workloads need to be scheduled in an efficient manner. Various scheduling algorithms are used by load balancers to determine which backend server to send a request to. It is that selected server which allocates resources and schedules the job dynamically on some virtual machine (VM) located on the same physical machine. Service provider is responsible to dynamically reallocate or migrate the VM across physical machines for workload consolidation and to avoid over-utilization or under-utilization of resources. Matching QoS requirements with a cost-effective amount of resources is challenging as workloads, user demands take large swings over time. Prediction is necessary as the virtual resources that cloud uses have their non-negligible setup time. A proactive dynamic provisioning of resources, that estimate the future need of applications in terms of resources and allocate them in advance, releasing them once they are not required, will be helpful.

Keywords: Cloud computing, Load balancing**,**Multi-resource optimization, Cache system, Reconfiguration cost, Cloud Partitioning, Workload Prediction.

I. INTRODUCTION

Since 2009, few IT developments have received more media coverage than cloud computing. Cloud computing is a new method for delivering business and IT services. It enables businesses and users to acquire IT resources they need, when they need it. It is a form of on-demand IT service provisioning and utilization. Cloud computing is the use of network resources, principally internet resources, to provide on-demand data processing and storage. As with many technology concepts, cloud computing has often been presented as a silver bullet that can solve companies' IT needs. Distributed computing and Grid computing have paved way for this new virtualized type of computing called cloud computing, that promises to chop the marketing and working costs and allowing on-demand computing.Cloud computing is considered as a sort of an application as well as a platform for the development. The applications which are developed are accessed over the internet which infers use of large data servers to host and run various web applications and services [1]. On the other hand while acting as a platform, it supplies, configures and re-configures the servers. Thus, cloud computing supports data and computational outsourcing, having:

- Unlimited resource scalability and elasticity.
- On-demand provisioning and allocation of resources.
- No extra cost incurred.

Virtualization is the primary technology behind cloud computing that permits the concurrent execution of different tasks over a shared infrastructure. Cloud gives computing resources in the form of virtual machine (an abstract machine that runs on physical machine). Since the co-located tasks do not interact with each other and access only their own data,it gives the user a feeling of working in an isolated environment. The isolation between each virtual machine is kept up by a Hypervisor, Virtual Machine Manager, which handles multiple operating systems by allocating needed resources and provides an identical abstraction of the underlying hardware to the virtual machines. As a scalable and distributed system, cloud computing requires allocation of the resources to several users. It is virtualization that gives the possibility to on-the-fly and on-demand configuration of physical machines to run different tasks, subsequently preventing resource wastage [3].
In spite of the fact that virtualization tries to balance the load of the whole system dynamically, there is always a possibility of over-utilization or under-utilization of resources. Overloaded servers lead to performance degradation whereas under-loaded servers cause poor utilization of resources. Due to inefficient distribution of load,the overloaded servers will generate more heatwhich in turn increase the expense of cooling system and substantial emission of CO_2 adding to greenhouse effect. On the other sidethe underutilized servers increase

power utilization of the overall system which is not under any condition environment friendly and in addition it builds on to the operational expenses. So it is must to provision right amount of resources dynamically to the applications running in virtual machines with a specific goal to meet the QoS requirement as well as to balance overall load of the system. Load balancing has become one of the major challenges in cloud computing. It is a mechanism that distributes the dynamic workload uniformly across all the nodes in the whole cloud to avoid a circumstance where some nodes are heavily loaded while others are doing no work or little work.The load refers to the work needed to be done on the system. By optimal resource utilization it helps to achieve less response time, high throughput,scalability, improved fault tolerance, high user satisfaction, less heat generation, optimum power consumption, reduced emission of CO_2 and less operational cost.The main aim is to serve the user's request by the efficient and effective utilization of the resources available in the system at the time.

The Cloud phenomenon brings along the cost-saving benefit of dynamic scaling.Static allocation of resources becomes ineffective, as amid a time of low demand there will be overabundance of resources available, bringing about pointless cost for the application provider, while amid high utilization time the available resources may be inadequate, leading to poor QoS and loss of costumers and revenue. Clouds go around this issue by enabling dynamic provisioning of resources to applications based on workload behaviour patterns, such as request arrival rate and service time distributions. To exploit dynamic application resource scaling, accurate decisions need to be made on when to scale resources up and down. This means that additional resources can be allocated for peak periods and can be released during the low demand periods, based on workload prediction by tracing the previous pattern, thus increasing utilization of deployed resources and minimizing the investment in Cloud resources without loss of QoS to end users[1].

II. CLOUD COMPUTING

Cloud computing is characterized as a service to distribute the computing power and storage capacity to the end users of a heterogeneous framework. Itrefers to the applications and the services running on the distributed network using virtualized resources and retrieved by common Internet protocols and networking standards.It is distinguished by the belief that resources are virtual and infinite and abstraction is applied for users on the details of the software supporting the physical systems.The fundamental idea inspiring the cloud computing is the handling, management and scheduling computing resources connected by a network and supporting user resources and services in correspondence evenly to the needs with payment based usage [2].

"Cloud computing is defined as a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resource that can be rapidly provisioned and released with minimal management effort or service provider interaction."

A. Characteristics

a. On demand self-service: Without requiring human interaction, users can obtain, use and release the cloud computing resources, mostly done through a web-based self-service panel.

b. Broad network access: All the cloud computing services are accessible over the network, supporting various client platforms.

c. Resource Pooling: The computing resources of a provider are pooled together to serve multiple users using multi-tenant model, with dynamic provisioning of different physical and virtual resources on user demand.

d. Rapid elasticity: It is the seamless scaling out and scaling in of the cloud resources. Services are provisioned and released as per the need of the user. They seems as infinite to a user.

e. Measured service: It reflects the pay-as-you-go model of utility computing. Provides accounting service of a resource involving monitoring, measuring and billing transparently based on utilization.

B. Framework

Cloud Computing is mainly modelled in two different ways, namely:

- Service Model: It includes different types of services that a user can access on a cloud computing platform.
- Deployment Model: It refers to the supervision, management and location in the cloud infrastructure.

The services comprising Service model are as below:

Fig. 1 Service Model layers.

1. Software As A Service(SAAS):

It is a complete software offering running at the top layer. It uses the web as the interface to deliver applications that are managed by cloud service providers and the interface is accessed on the user's side on a paid basis. Because of this, SaaS eliminates the need of having installed and running applications on individual systems [5].

2. Platform As A Service (PAAS):

At the middle layer, it provides a platform to develop and deploy software. PAAS abstracts operating system, server hardware and software, and network infrastructure, leaves user to work on application development for their project.

3. Infrastructure As A Service:

To the bottom of the stack, the fundamental building block for cloud services, i.e. IAAS comprises highly automated and scalable compute resources, with cloud storage and network capability that are self-provisioned, metered and available on-demand. Instead of purchasing absolute hardware, users can purchase IAAS based on consumption.

The cloud is made on the following deployment model:

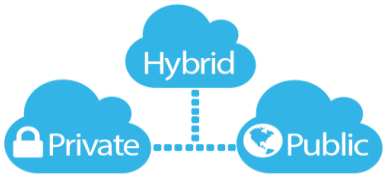

Fig. 2 Types of deployment models.

1. Public Cloud:

These are the clouds made available to the general public by a service provider who makes resources, such as applications and storage. Users have no visibility or control over where the infrastructure is located and all customers share the same infrastructure pool.

2. Private Cloud:

These are the proprietary network or data center that uses cloud computing technologies. They can be: On-premise clouds (hosted within an organizations own facility) and externally hosted private clouds (exclusively used by one organization, but hosted by a third party.

3. Community Cloud:

These are multi-tenant cloud service model that is shared among several organisations sharing similar needs or goals. They are governed, managed and secured either by all involved organisations or a third party service provider.

4. Hybrid Cloud:

These are a composition of two or more clouds (private, public or community) that remains as individual entities but are bound together offering the advantage of multiple deployment models. They also provides the opportunity to store sensitive and critical information in a public cloud, and non-critical in the public cloud. These helps in providing varying levels of security, control and scalability.

III. LOAD BALANCING IN CLOUD ENVIRONMENT

Load Balancing is a method to distribute workload evenly across all the nodes through network links to achieve optimal resource utilization for maximizing throughput and minimizing overall response time.Load balancing is a process of scattering the load among a number of resources in a system. Thus in cloud-based architecture,load need to be distributed over the resources, so that each resource does approximately the equal amount of work at any point of time. It is used for avoiding too much overload on the resources and dividing the traffic between servers. Data can be sent and received at less delay. Load Balancing is used for minimizing the total waiting time of the resources. In cloud computing it is used in balancing the load on virtual machine and cloud resources.

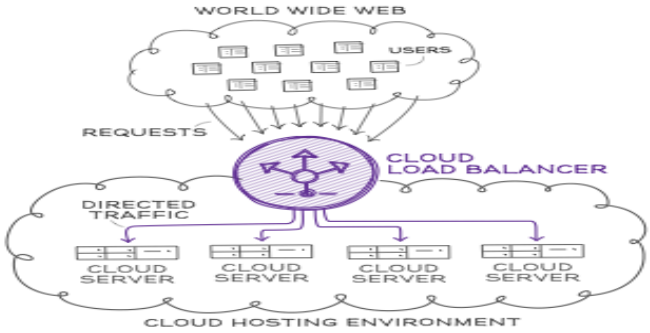

Fig. 3 Load balancing in a cloud computing environment.

A. Goals of Load Balancing

- To maintain the system stability.
- To improve the performance significantly.
- To increase the flexibility of the system.
- To accommodate future modification in the system.
- To have a backup plan in case the system fails even partially.

B. Types of Load Balancing algorithms

On the basis of the initiator of the process, it can be classified as:

- Sender Initiated: Case when the load balancing algorithm is initiated by the sender. A heavily loaded node will initiate this by requesting the present load statistics from other nodes and then assigning the task to the lightly loaded node.
- Receiver Initiated:Case when the load balancing algorithm is initiated by the receiver. A lightly loaded node looks for heavily loaded nodes from which it may receive task.
- Symmetric: Combination of both sender initiated as well as receiver initiated techniques to work at different load conditions.

On the basis of current state of the system, it can be classified as:

- Static Load Balancing:

In static load balancing algorithms, the execution of the processors is decided at the start of the execution, and no dynamic statistics are used. It means that these algorithms don't depend on the current state of the system. For this, prior information of the system is required. It becomes a drawback in some cases when the complete information needed is not available at the allocation time. They suits more to a homogenous and stable environment and can produce higher results.
Static load balancing algorithms are non-preemptive, thus an assigned process to a node cannot be changed during process execution. They aim at minimizing the execution time of a task and reduce communication delay and overhead.

- Dynamic Load Balancing:

In dynamic load balancing algorithms, the decisions are based on the current state of the system. They take into considerationthe different attributes of the nodes' capabilities and network bandwidth. They mainly rely on the combination of prior gathered information about the nodes and run-time information collected as the nodes execute task. These algorithms can assign and reassign the tasks dynamically to the nodes based on attributes collected and calculated. It requires constant monitoring of the nodes and task progress.
Dynamic load balancing algorithms allows a process to move from an over utilized node to underutilized node dynamically. This means that they supports process pre-emption. One of the important point in using dynamic load balancing is that if some node fails, it will not halt the system, instead it will only affect the performance of the system. Dynamic algorithms are more versatile than static algorithms, can easily adjust to the modifications and give better results in heterogeneous and dynamic conditions. Dynamic load balancers keep track of updated attributes through policies, as: Selection policy, Transfer policy, Location policy and information policy.

IV. LITERATURE SURVEY OF LOAD BALANCING ALGORITHMS

The lifetime of a system is directly affected by the efficiency of the load balancing algorithms it has implemented. The load balancing algorithms are designed to balance the load in the system for producing the optimized results as a whole. The datacentre controller handles the user submitted tasks, which in turn uses VmLoadBalancer to determine the VM to assign the next job. The VmLoadBalancer may use the following algorithms for load balancing:

A. Round Robin Algorithm

Round Robin is one of the easiest scheduling techniques that works on the concept of time slices. In this, the time is divided into multiple slices and each node is given a specific time slice or time interval i.e. it uses the time-shared scheduling strategy. The node will perform its task in the time quantum allotted to it. The client gets the needed resources of the service provideron the basis of this time slice. The following figure shows the working of round robin. It shows that each user request is served by every processor within a given time interval. After the time slice gets over, the next user request in queue will get the processor. In case of the completion of request within time quantum, user does not need to wait, otherwise user have to wait for its next slot to get served.

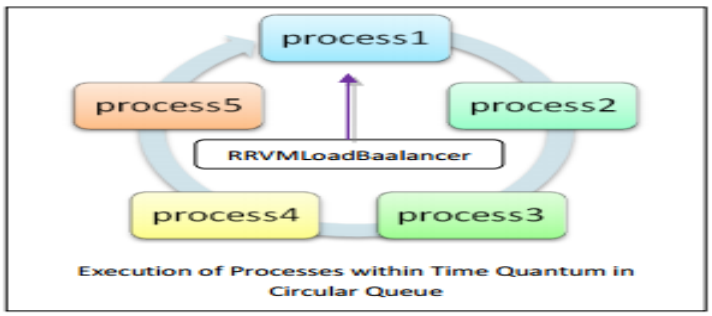

Fig. 4 Round Robin processing.

AVM is randomly picked from the pool to serve the first request and afterwards the requests are assigned in a circular manner. Once the VM is assigned a request, it gets listed down to the end of the list. In spite of the fact that the work load is equally distributed among processors but the execution time for different processes differ. This means that at any given point of time some nodes may be heavily stacked and others remain idle.
RRVLB: RoundRobinVmLoadBalancer

Step-1:
Round RobinVmload Balancer maintains a data structure which comprises of an index of VMs and current Vmsstate (i.e. busy/available).
Initially all vm'sare free
Step-2:
a. The datacentercontroller receives the user requests/cloudlets.
b. It stores the arrival time & burst time oftheuser requests.
c. The requests allocation to Vmsis done on the basisof their state information from the VM queue.
d. The RRVLB will allocatethe time quantum for user request execution.

Step-3:
a. The RRVLB will calculate the turn- around timeof each process.
b. It also calculates the response time and average waiting time of user requests.
c. It decides the scheduling order.
Step-4:
After the execution of cloudlets, the Deallocation of VMs takes place by the algorithm.
Step-5:
The datacentercontroller checks for new/pending/waiting requests in queue.
Step-6:
Continue from step-2.

B. Throttled Algorithm

Throttled is completely a virtual machine based algorithm.The client first send request tothe load balancer to find a suitable Virtual Machine to perform the specified task. It starts by maintaining a list of all the VMs; each row is separately indexed to speed up the lookup process recording the state of virtual machine that can be either busy or idle. Load balancer will accept the user request and allocates the matched VM to serve the request on finding a match based on size and availability. In case, if no VM is available that can match the needed criteria, the load balancer returns -1 and the request is queued.

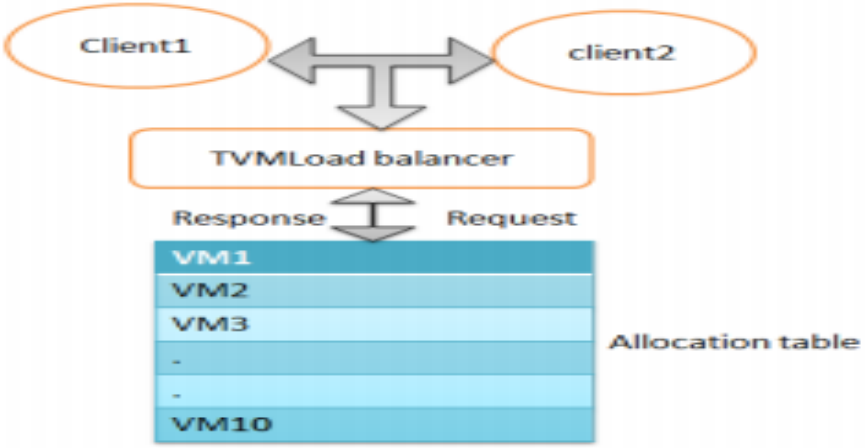

Fig. 5 Throttled processing.

TVLB: ThrottledVmLoadBalancer

Step-1:

TVLB maintains vector which comprise of fields: index VMs and thestate ofthe VM(i.e Available or Busy).

Initially all VM'sare free i.e. available for serving request.

Step-2:

DataCenterController receives a new request.

Step-3:

DataCenterControIler queries the TVLB for thenextallocation.

Step-4:

TVLBscanthe entire vector until the firstavailable VM found.

If found:

i) The TVLB return the VM id to the DataCenterController.

ii) The DataCenterController sends therequest to the identified VM.

iii) DataCenterController notifies the TVLB of the new allocation.

iv) TVLB updates the allocation table accordingly.

else

i) The TVLB returns -1.

ii) The DataCenterController queues therequest.

Step-5:

After VM finishes processing therequest, and the DataCenterControllerreceives the responsecloudlet, itnotifies the TVLB of the VM de-allocation.

Step-6:

The DataCenterController checks if there areany pending i.e. waiting requests in the queue. If thereare, then continuesfrom step3.

Step-7:

Continue from step 2

C. Equally Spread Current Execution (ESCE) Algorithm

ESCE is a spread spectrum strategy in which the load balancer spread the load in hand into multiple virtual machines. A queue is maintained by the load balancer of the jobs that requires and are currently utilizing the services of the virtual machine. The load balancer then persistently examines this queue and the list of virtual machines. In the event that there is a VM available that can operate on the request of the node/client, then that VM is allocated to that request. However, if there is a VM that is free and another VM needs to be freed of the load, then the balancer distributes a percentage of the tasks of the loaded VM to the free one so as to reduce the

overhead of the loaded VM. The jobs are submitted to the VM manager, the load balancer also maintains a list of the jobs, their size and resources requested. The load balancer selects the job that an available VM can perform at the present time.The balancer works the way so as to improve the response time and processing time of a job by selecting it whenever it get a match.

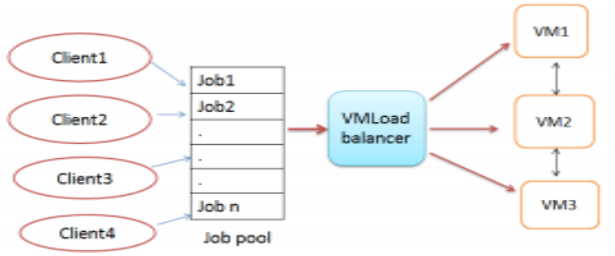

Fig. 6 Equally Spread Current Execution processing.

Algorithm
Step-1:
Select the next VM available in the pool
Step-2:
Check for all current allocation count <max length of VM list allocate the VM
Step-3:
If VM<>available then launch a new vm instance
Step-4:
Note the active load on each VM instances
Step-5:
Return_ids(least_loaded_vm(s))
Step-6:
TheVMLoadBalancer will allocate the requesttoone ofthe VM from the least loaded vm(s)
Step-7:
If a VM is overloaded then theVMLoadBalancerwill distribute some of its work to the VM having least work so
that everyVM is similarly loaded.
Step-8:
The DataCenterController receives the responsetothe request sent and then allocate the waitingrequestsfrom the jobpool to thevacant VMs so on.
Step-9:
Continue from step-2.

V. LOAD BALANCING: A DIFFERENT PERSPECTIVE

A. Distributed Cache system

Recently, distributed objects which are highly scalable are used for speeding up applications by avoiding accesses to database. Objects are assigned to hashing based cache instances, and they are shifted from one cache instance to another only when more number of instances are added to the cache and the objects are needed to redistribute. This may prompt circumstances where some cache instances get overloaded by the requests asthey store some objects that are frequently accessed, leaving other cache instancesless frequently utilized compared to them.

A multi-resource based load balancing algorithm for distributed cache systems aims at balancing both memory and CPU among cache instances by repositioningcached data. Depending on the runtime load distributions, weighted priorities are associated with memory and CPU resources in order to avoid conflict while balancing.Scarcer the resource is, more is its weight when load balancing. The load balancing algorithm will make some data partitions to move from one cache node to another for rebalancing the load. This migration of data partitions also incurs cost in terms of someconsumption of resources such as CPU and network bandwidth which makes reconfiguration cost to betaken into account by the load balancing algorithm. The greedy algorithm approach expects a local efficient solution to minimize the system imbalance degree in each and every step, while a nearest-neighbour-search based policy is based on selecting the bucket i.e. data to be migrated.

The system imbalance degree is calculated based on the utility load of a node as well as monitoring information. Moreover, since continuous rebalancing of the system may degrade the Quality of service of applications

utilizing the cache system, each data migration should minimizes the system imbalance degree which in turn minimizes the total reconfiguration cost.

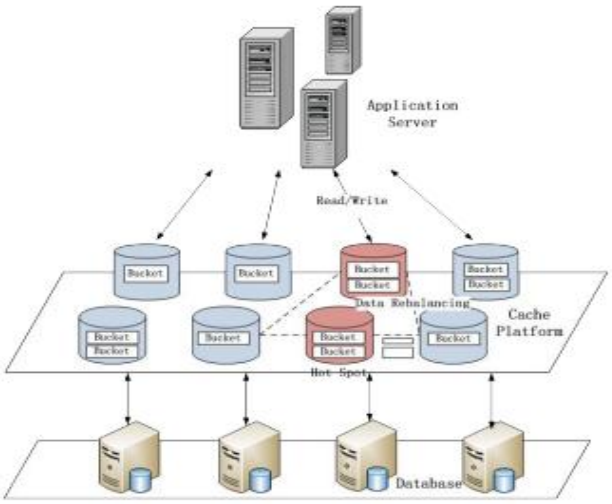

Fig. 7 Load Balancing scenario in a Cache Platform.

Considering that cache systems is having limited resources both CPU and Memory, the load balancing algorithm must endeavour to balance both resources to take out hot spots. The aim is to achieve the following goals:

1) Lessening the load imbalance degree for multiple resources (CPU and Memory) among cache nodes with resources having different priorities based on the runtime load assessment.

2) Lessening the system reconfiguration cost, in terms of the number of data buckets to be migrated.

Definition	Description
$N = \{n_1, n_2, \dots, n_k\}$	Set of cache nodes, k is the number of cache nodes in system.
$Q = \{q_1, q_2, \dots q_i, \dots q_k\}$	Number of buckets in each node. q_i is the number of buckets in node n_i.
$B_i = \{b_{i1}, b_{i2}, \dots b_{iq_i}\}$	Set of buckets in node n_i.
$NL = \{(c_1, m_1), (c_2, m_2) \dots (c_k, m_k)\}$	Resource usage at each node, (c_i, m_i) is the average (CPU, Memory) resource usage in % at node n_i in a time interval.

The load of each bucket depends on its storage size and its CPU consumption. The pair (c_i, m_i) as (CPU, Memory) defines usage of cache node i. The utility node is defined as:

$$l_i = \lambda * c_i + (1 - \lambda) * m_i$$
$$\lambda = (\sum_{i=1}^{k} c_i) / \sum_{i=1}^{k} (c_i + m_i)$$

In order to capture runtime load of a node, dynamic weights λ and 1- λ are used for CPU and Memory. σ_{cpu} and σ_{mem} evaluates imbalance degree of CPU and Memory, σcluster captures cluster's imbalance degree.

$$\sigma_{cluster} = \sqrt{\mu\sigma_{cpu}^2 + (1-\mu)\sigma_{mem}^2}$$
$$\sigma_{cpu} = \sqrt{\sum_{i=1}^{k} (c_i - \overline{c})^2 / k}$$
$$\sigma_{mem} = \sqrt{\sum_{i=1}^{k} (m_i - \overline{m})^2 / k}$$
$$\mu = \lambda^2 / (\lambda^2 + (1-\lambda)^2)$$

The algorithm migrates a bucket from heaviest utility load node to the lightest utility load node. The nearest-neighbour search policy which selects the bucket to be migrated takes its decision as:

Let at the current state, system imbalance degree is $\sigma_{cluster}$, n_a and n_b are the most loaded and lowest loaded node. If after moving the bucket system imbalance degree becomes $\sigma^*_{cluster}$, then it will move if:
(1) $\sigma^*_{cluster} < \sigma_{cluster}$
(2) $\sigma^*_{cluster}$ is minimal.
Given a point h, target bucket to move should be the one closest to this point.

$$h: (\lambda (c_a - c_b)/2, (1-\lambda)(m_a - m_b)/2)$$

B. Cloud Partitioning

This model is aimed at the public cloud which has innumerable nodes having distributed computing resources scattered among different geographic locations. Cloud partitioning comes into account to manage this. A cloud partition is basically a subarea of the public cloud divided according to geographic locations. The cloud (geographically partitioned) has a main controller that chooses the suitable partitions for arriving jobs while the balancer for each cloud partition chooses the best load balancing strategy.
When a job arrives on the server, the main controller queries the local cloud partition where the job is located. If the status of this location is idle or normal, the job is handled locally. If not, then another cloud partition is found that is not overloaded.
The load degree of each node are entered into the Load Status Table created by cloud partition balancers which refreshes after a fixed interval of time [18].

VI. CONCLUSION

Cloud Computing has been widely adopted, though there are many issues with it. Load balancing is one of the primary issue in cloud computing. It is required to distribute the workload evenly across all the nodes to achieve a high user-end satisfaction and optimum resource utilization. Because when a client request a service, it need to be available. When a node is overloaded with job, a load balancer has to set that load on other free nodes, so as to balance the system degree.
In multi-resource scenarios, it is required to balance resources in respect to the weight associated to a resource. For a distributed cache system, in order to improve the system imbalance degree, data partitions are migrated depending on the nearest-neighbor search algorithm.

VII. REFERENCES

[1] A Multi-Resource Load Balancing Algorithm for Cloud Cache Systems: Yu Jia1, Ivan Brondino, Ricardo Jiménez Peris, Marta PatiñoMartínez , Dianfu Ma
[2] Load Balancing and Resource Monitoring in Cloud:Nitin S. More
[3] Proposal for an Optimal Job Allocation Method for Data-intensive Applications based on Multiple Costs Balancing in a Hybrid Cloud Environment:Yumiko Kasae,Masato Oguchi
[4] A Load Balancing Model Based on Cloud Partitioning for the Public Cloud GaochaoXu, Junjie Pang, and Xiaodong Fu.
[5] A COMPARATIVE SURVEY ON LOAD BALANCING ALGORITHMS IN CLOUD COMPUTING Hamid ShojaHosseinNahid Reza Azizi.
[6] A Comparison of Four Popular Heuristics for Load Balancing of Virtual Machines in Cloud Computing SubasishMohapatra, K.SmrutiRekha,SubhadarshiniMohanty
[7] A Survey on Scheduling and Load Balancing Techniques in Cloud Computing Environment: Subhadra Bose Shaw, Dr. A.K. Singh
[8] A Survey On Load Balancing In Cloud Computing Ms.Parin. V. Patel, Mr. Hitesh. D. Patel , Asst. Prof.Pinal. J. Patel.
[9] A Survey Of Various Load Balancing Techniques And Challenges In Cloud Computing: Tushar Desai, JigneshPrajapati.
[10] A Comparative Study of Load Balancing Algorithms in Cloud Computing Environment MayankaKatyal, Atul Mishra.
[11] Round Robin Approach for VM Load Balancing Algorithm in Cloud Computing Environment Nusrat Pasha, Dr. Amit Agarwal,Dr. Ravi Rastogi.
[12] Execution analysis of load balancing algorithm in cloud evnvironment:Soumya Ray and Ajanta De Sarkar.
[13] Load Balancing On Cloud Data Centres:Dr.Hemant S. Mahalle, Prof.Parag R. Kaveri, Dr.VinayChavan.
[14] Load Balancing in Cloud Computing:Rajwinder Kaur and PawanLuthra.
[15] An Analysis of Load Balancing in Cloud Computing: Suresh M., ShafiUllah Z., Santhosh Kumar B.
[16] Design of an Optimized Virtual Server for Efficient Management of Cloud Load in Multiple Cloud Environments: Ajay A. Jaiswal1, Dr. S. K. Shriwastava.
[17] The Load Balancing Algorithm in Cloud Computing Environment: HaozhengRen, YihuaLan, Chao Yin.
[18] A Survey on Load Balancing in Cloud Computing Using Soft Computing Technique's: Aloksingh ,Vikas Kumar Tiwari , Dr.BhupeshGour.
[19] An Analysis of the Load Scheduling Algorithms in the Cloud Computing Environment: A Survey: Divya Chaudhary, Bijendra Kumar.
[20] Analytical Study of Load Scheduling Algorithms in Cloud Computing: Divya Chaudhary, Bijendra Kumar. Bhatia, Jitendra.; Patel, T.; Trivedi, H.; Majmudar, V., "HTV Dynamic Load Balancing Algorithm for Virtual Machine Instances in Cloud," Cloud and Services Computing (ISCOS), 2012 International Symposium on , vol., no., pp.15,20, 17-18 Dec. 2012.