Load Balancing In Cloud Computing

[1]Akash Jain, [2]Ms.Pinal Patel
[1]IT System & Network Security
[1]Gujarat Technological University, Ahmedabad, India

Abstract: **Cloud computing can be define as a structured model that which defines computing services, in which resources as well as data are retrieve from cloud service supplier via internet through some well shaped web-based device and application. It provides the on demand services for various applications and infrastructure to the user. Cloud service providers are required to provide the service efficiently and effectively. For that, a cloud provider utilizes all the resource from the node. Thus, the node that are meant for creating a task in the cloud computing must be considered for efficient usage of the available resources. Resources have to be properly selected according to the properties of the task. By analyzing the present research on cloud computing, we have come to the most common and important issue of load balancing. Load balancing has been always a study topic whose purpose is to make sure that all computing resources are circulated proficiently and fairly. As numbers of users are increasing on the cloud, the load balancing has become the challenge for the cloud provider. Load balancing being subject of research, proposed algorithm for load balancing which will work dynamically for optimal usage of resource utilization.**

Index Terms **– Load balancing, Cloud Computing, Load balancing technique , User base priority.**

I. INTRODUCTION

Cloud computing is computing in which large groups of servers are networked to allow central data storage and online access to computer services or resources. Clouds can be classified as public, private or hybrid. Cloud computing relies on restrict to sharing of resources to achieve consistency and economy of scale, similar to a usefulness over a network. At the foundation of cloud computing is the broader concept of converge infrastructure and shared services. Cloud computing, or in simpler shorthand just "the cloud", also focuses on maximizing the efficiency of the shared resources. Cloud resources are usually not only shared by multiple users but are also animatedly reallocated per demand. This approach should maximize the use of computing power thus dipping environmental damage as well since less power, air conditioning, rack space, etc. are required for a variety of functions. With cloud computing, multiple users can access a single server to recover and update their data without purchasing licenses for different applications. Proponents claim that cloud computing allows companies to avoid upfront infrastructure costs, and focus on projects that differentiate their businesses instead of on infrastructure. Proponents also claim that cloud computing allows enterprises to get their applications up and running faster, with improved manageability and less maintenance, and enables IT to more rapidly adjust resources to meet fluctuating and unpredictable business demand. Cloud provider typically use a "pay as you go" model. This can lead to suddenly high charges if administrators do not adapt to the cloud pricing model.

II. INTRODUCTION TO LOAD BALANCING

In computing, **load balancing** distributes workloads across multiple computing resources, such as computers, a computer cluster, net links, middle processing units or disk drives. Load balancing aims to optimize resource use, maximize throughput, minimize response time, and avoid overload of any single resource. Using multiple components with load balancing instead of a single part may add to consistency through redundancy. Load balancing generally involves dedicated software or hardware, such as a multilayer key or a Domain Name System server process.

Load balancing is the process of improving the presentation of the system by shifting of workload among the processors. Workload of a machine means the total processing time it require to execute all the tasks assigned to the machine [5]. Load balancing is done so that every virtual machine in the cloud system does the same amount of work throughout therefore rising the throughput and reduce the response time. Load balancing is one of the main factors to intensify the working performance of the cloud service provider. Balancing the load of virtual machines consistently means that anyone of the existing machine is not idle or partially loaded while others are heavily loaded. One of the critical issue of cloud computing is to split the workload dynamically.

III. WHY LOAD BALANCING IN CLOUD?[15]

Load balancing in clouds is a device that distributes the overload dynamic local workload evenly across all the nodes. It is used to achieve a high user happiness and resource consumption .ratio , making sure that no single node is besieged, hence. Improving the overall performance of the system. Proper load balancing can help in utilize the available resources optimally, there by reduce the resource consumption. It also helps in implementing fail-over, enable scalability, avoiding bottlenecks. and over-provisioning, reducing response time etc. The factors responsible for it are:

- Limited Energy Consumption: Load balancing can reduce the amount of energy spending by avoiding over hearting of nodes or virtual machines due to extreme workload .

- Reducing Carbon Emission: Energy use and carbon emission are the two side of the same coin. Both are directly relative to each other. Load balancing helps in reducing energy use which will automatically reduce carbon production and thus achieve Green Computing[6].

Load Balancing :The goals of load balancing are:

- Develop the performance considerably
- Having a backup plan in case the system fail even partly
- To continue the system constancy
- To put up future change in the system.

Analysis of Load Balancing Algorithm *[1]*

These are the categories of load balancing algorithm

With two initiated the process, load balancing algorithms can be of three categories:

- Sender Initiated: If the load balancing algorithm is initialized by the sender, it is called sender initiated.
- Receiver Initiated: If the load balancing algorithm is initiated by the receiver,then it is called receiver initiated.
- Symmetric: It is the mixture of both sender initiate and receiver initiate Depending on the current state of the system, load balancing algorithms can be divided into two categories:

Static Algorithm:

Static algorithm divide traffic regularly between the servers. By this method the traffic on the server will be disdain easily and therefore it will make the situation more incorrectly. 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 serious challenge linked with round robin.

Round Robin Algorithm

Round Robin algorithm distributes job evenly to all slave processors. All jobs are assigned to slave processors according to Round Robin order, sense that processor choosing is performed in sequence and will be back to the first processor if the last processor, free of allocation of other processors.

Randomized Algorithm

Randomized algorithm uses casual numbers to choose slave processors. The slave processors are chosen randomly following arbitrary information generate based on a statistic distribution.

Central Manager Algorithm

Central processor will select a slave processor to be assign a job. The selected slave processor is the processor having the smallest amount load. The middle processor is able to gather all slave processors load information, thereof the choosing based on this algorithm are possible to be performed. The load manager makes load balancing decision based on the system load information, allowing the best decision when of the process produced.

Threshold Algorithm

In this algorithm, the processes are assign right away upon creation to hosts. Hosts for new processes are chosen locally without sending isolated messages. Each processor keep a personal copy of the system's load. The load of a processor can characterize by one of the three levels: Under loaded, medium and congested. Two entrance parameters t_under and t_upper can be used to describe these levels. **Under loaded: load < t_under, Medium : t_under** ≤ load ≤ t_upper**, Overloaded: load > t_upper.**

Dynamic Algorithm

This algorithms elected proper weights on servers and by probing in whole network a lightest server chosen to balance the traffic. However, selecting an appropriate server needed real time contact with the networks, which will lead to extra traffic added on system. Dynamic algorithm predicated on query that can be made regularly on servers, but sometimes prevail traffic will stop these queries to be answered, and also more added overhead can be famous on network.

Central Queue

It stores new actions and unfulfilled requests as a cyclic FIFO line on the main host. while new activity arrives in the queue manager, it will inserted into the queue. Then, when a request for an activity is received by the queue manager, the queue manager select the first activity from the queue and sends it to the requester. If there are no ready activities in the queue, the request is buffer, until a new activity is available. If a new activity arrives at the queue manager while there are unanswered requests in the queue, the first such request will send from the queue and the new activity is assigned to it.

Local Queue

The basic idea of the local queue algorithm is static allocation of all new processes with process migration initiated by a host when its load falls under threshold limit, is a user-defined parameter of the algorithm. The parameter defines the minimal number of ready processes the load manager attempts to provide on each processor.

Load Balancing With Cost Scheduling Algorithm.[3]

The basic working of this simplest form a cloud user connects to the cloud via a cloud provider/server or a cloud broker . The user submits his request for resource to the cloud through the cloud provider. The cloud provider assures optimal efficiency. To provide better service to the user it applies the optimization algorithms.

The request is actually executed at the cloud using virtual machines and deploying the available pool of resources. These are available as the cloud middleware. The resources that are available as services are storage service, network service or operating system service.

It shows how the load balancer distributes the load among the different VMs/virtual server so that the processing of the request is completed and the user gets the service. As the load IS processed among all the VMs, no VM is over loaded.

By using the above notations, we define the cost as C. The cost of execution depends on the package Pkg containing the resource R executing on virtual machine VM.

Table:1 key variables used

Variables	**Meaning**
Ri	The available cloud resources
VMi	The available virtual machines
Ci	The Price fixed for VMi executing Ri
u_cost	The cost of the user for getting the service
e_cost	The cost taken by VM to serve the user
u_time	User waiting time
Pri	Profit at provider for executing the resource
I	Number of instances, ranging from 1 to n
P	Processor
Pkg	Resource grouped into packages

LetCi = (VMi,Pkg,Ri)

The algorithm works as follows.

```
if (P[i]==O) [Check the processor status , if it is idle]
Processor status= 1; [Set processor status as ready]
u cost=u cost[i]; [Initialize user cost]
u_time=u_time[i]; [Initialize user time]
e_cost = (Time taken by VM *U_time); [Calculate the actual cost]
Pr = u_cost - e_cost; [Calculate the profit]
```

By using this algorithm both cloud provider and cloud users are benefited.

The profit at the service provider can be calculated as:

u_cost = ∑u_cost (VM)/hour [user pays according to the package selected]

e_cost = ∑e_cost (VM)/hour [actual cost involve inexecuting]

A service request for an application consisting of one or more services is sent to a provider specifying two main constraints, time and cost. Cloud Users first register themselves in cloud data center. Once they are registered they can avail the desired services. Once the user logs in, a list of services available at the provider along with the cost is displayed.

Based on the requirement the user can select the package and the time duration he wants the service. Some packages are made available for the cloud user. Based on their need they can choose the package that they can afford.

HTV Dynamic Load Balancing Algorithm for Virtual Machine Instances in Cloud[4]

On deep studies, some of the important troubles of cloud were examine, like data security, data loss, load balancing etc. In cloud environment, resource allocation takes place majority at two levels.

1st level: Whenever an application is uploaded to the cloud, the load balancer assign the requested instance to physical computers and try to attempting to balance the computational load of applications across physical computers.

2nd level: When an application receives incoming requests, each requests must be assign to a exact application instance to balance the computational load across a set of instances of the application.

This will grab the performance of heavy loaded node. If some good load balancing technique is implemented, it will equally divide the load and thereby increase resource utilization.

Round Robin workings well in most configurations, but it could be more efficient if the tools that load balancing is about equal in processing speed, connection speed, and memory. The basic idea of working behind the algorithm is that the system generates a standard round queue and walks through it, sending one request to each machine before it will getting to the start of the queue and repeating that queue again. The problem with this approach is that it will not check the server is heavy loaded or not, it will directly assign the request whenever its turn comes. So that is the reason.

Fig. depicts the proposed working model on which the research is to be carried out. This model provides us the platform to implement the research on load balancing within cloud computing. The basic components involved in this architecture are Cloud Controller, Node Controller, Virtual machines, User and Load Balancer.

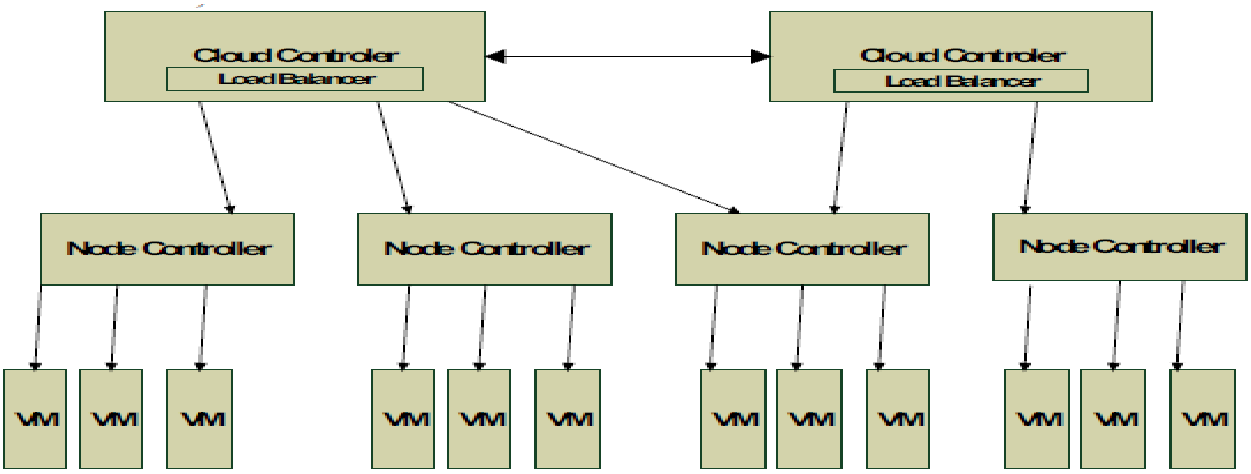

Figure 1 Existing Architecture

Cloud Controller (CL): The first phase of cloud infrastructure is cloud controller. It is the root of overall structure. The request sent by a user is authentically accepted by the cloud controller and passed to next phase of this structure. This phase provides a web interface to user and also interacts with the rest components of cloud infrastructure. Load balancer algorithm would be implemented on cloud controller.

Node Controller (NC): The next phase is Node Controller. The request accepted by cloud controller is passed to node controller. It is a Virtual extension (VT) enabled server capable of running KVM (Kernel Virtual Machine) as the hypervisor. The entire process is controlled by cloud server. The VMs running on the NC are called instances. Node Control Server runs on each node and controls the life span of instances running on the node. The NC interacts with the OS and the hypervisor running on the node along with the Cloud Control.

Virtual Machines (VMs): VMs can be considered as instances of the cloud. Different types of instances are created for every user depending upon the demand of services. The request of the user is fulfilled through instances. Instances are stored within Node Controller. The Load balancing of these instances are done as the request for service is received from the client. User: Users are generally the cloud users who demands for services and have access to those services of the cloud.

Load Balancer: It is a new load balancer algorithm implemented within cloud controller which would balance the load as per the user request.

IV.IV EXISTING SYSTEM[4]

Below is the scenario of how the algorithm is actually works. Depending upon the whole research, the concept can be implemented through the following algorithm.

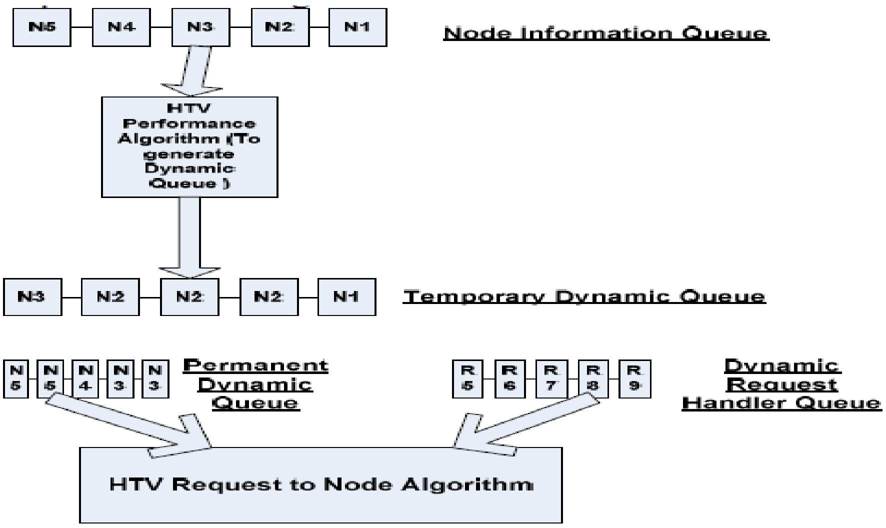

Figure 2: Existing working model

There are various steps:

Node information queue : It will have all the information about all nodes contain the parameter like basic node information, free space and performance details. These information will be gather by sending a request to cloud node and information about the resources will be communicate as a response from the node.

HTV performance algorithm: It will compute the information like performance, load on the particular node, total space available on the node. It will store these kind of information in the queue. Using these information, the load balancer will work dynamically. The load balancer works for load balancing of resources which in turn will help in proper allocation and distribution of resources. Generally, there are two parameter which are used namely load on nodes and response time of node.

Temporary dynamic queue: It will stores the information which are given by the HTV algorithm. It will store the information about the nodes list which will be more used for the allocation.

Permanent Dynamic queue : While queue is generated by HTV performance algorithm, permanent queue will be replace by temporary queue for next revolt of HTV performance algorithm. Every time when it will be updated after monitoring. thus if a new client request arrive it will be assigned to the current node pointer in permanent queue.

Working of HTV Algorithm: For better performance of this algorithm two parameters are taken in to consideration.

1. Load on the server.
2. Current performance of server.

Load on the server: We are more interested in freeresources available on node. The node having more free resources will able to handle more requests easily without degrading its performance.

Current performance of Sever: Current performance of the server is measured by sending request to the node at regular interval and while getting response the performance of the server will be measured. It might be happen the response time of node may changed every time. Thus, the above two parameters are considered to built a new queue for allocation. This information of the whole node is considered by a mathematical meaning to count z-parameter value for each node.

Existing Algorithm

```
Step 1: [ In this step Load Factor x will calculated]
x ← (Total _Resources – Used _Resources)
// where x will be the free memory in percentage.
Step 2: [In this step Performance Factor y will calculated]:
y1 ← average (current _response _time)
y ← y1 - (In this step it will find earlier calculated y1)
y ← y/(previous y1)*100
Step 3: [in this step it will find  z to find unavailable node] z ← x - y;
If (z < 0)z = 0;
Step 4: [This step will find minimum of all z except the nodes with z value 0]
Min _z= min (all z’s)
Step 5: [It will find Find min_ factor and divide every z by this factor]
Min _factor ← min _z
Z← z / min _factor
Step 6: [In this step it will generate Dynamic Queue on base of z]
```

In the algorithm x is measured as a free load on server, while is y for the performance on the server and y1 is for current response time of the server.

Explanation

Step-1: The value of x is generated by consider the total existing resources and allocated resources on the server. The available resources would be generated using the equation x = (Total _Resources – Used _Resources). While x value will calculated for all nodes servers we will find the available free load of the servers.

Step 2: The performance calculates the increase or decrease in performance on the server and the considered value is stored as y. Now for scheming y a request is send to all the nodes at regular interval of time and the response time is designed. So every hour sever will have a variety of values of y. By averaging all the value of y1 will be calculated to generate a queue. Now, the before considered y1 will be deduct from current values y1 which was currently used to evaluate the performance. The same way the increased or decreased performance is considered and the value of y will calculated as the percentage of earlier count y1. Which is y/(previous y1)*100.

Step 3: Including z = x - y; Here value of Y is subtract from X value to count the z value Here they are involved in the node with the lowest response time so, they take off the y value from x. i.e. nodes having extra response time will contain fewer z value and they will get less number of requests to handle. Assume in the bad case node have very less memory available and very large response time than z = x – y may get -ve value so they need to remove that node from queue and it will not think in any step of the algorithm and in this iteration of algorithm .

If any node is momentarily engaged the response time will be endless of that exact node. So the y value also become too large for that node which will lead to deficiency z value. And in step 4 of algorithm that node will useless for future process in this of algorithm. So with this pace we can also notice the nodes which are unavailable.

Step 4: While the z is calculated then the minimum of all z which we considered are stored in min _z. They will not consider node with 0 z value so, this node will be remove.Next iteration of the algorithm the z value of node will count again so it will get the then chance to join cloud environment.

Step 5: Here smallest amount factor is calculated and will divided by all the factors of z.

Suppose z values for node 1 to 5 are given below:

Node	**z-value**
S1	22.30
S2	12.23
S3	43.00
S4	14.36
S5	28.29

Now S2 has a list z value and so every nodes z value are divided by S2's z value and food a float. So now the z values are 2,1, 4, 1, and 2. Node with z value 0 will get 0 as z / min_fact so they can't get any request to switch.

Step 6: From the above value S1 has the capacity to handle two requests, node S2 can handle only one while S3 will handle 4 to keep load balanced on each node. So the temporary queue will look be prepared as follows:

S3	S3	S3	S3	S1	S1	S5	S5	S4	S2

Figure 3 : Dynamic Queue

So, once the provisional queue and lasting queue will be changed. First 4 requests will go to node 3 than 2 will go to S1 and so on until the end of queue. While the queue is over it will assign next 4 to S3 and go so on. In this way the whole algorithm will work.

V. PROBLEM WITH EXISTING SYSTEM

After studying previous algorithm "HTV load balancing algorithm of load balancing" following points could be considered : Existing models or suggested system architectures are implemented or designed for only two parameters.

1. Load on the server
2. Performance of the server

This algorithm was implemented only for two parameters for better load balancing in cloud scenario ,not consider the other parameters like resource utilization in terms of CPU utilization and RAM utilization ,user base priority, cost optimization etc..

Problem definition:

Keeping in mind the limitations of existing systems and models of HTV load balancing algorithm , I am proposing to implement user base priority parameter in it t which will contain following characteristics.

- User base priority:

***Advantage 1* -** A work on the share of resources in a dynamic cloud by using Priority algorithm which decides the allocation series for different jobs requested Among the dissimilar user after allowing for the priority based on some best threshold decided by the cloud owner.

***Advantage 2* -** This resource portion technique is more competent than others. With the arrival of cloud computing and by using this implemented priority algorithm the cloud admin can easily take decision based on the different parameters for executing the request.

VI. PROPOSED SYSTEM AND ALGORITHM

Proposed System

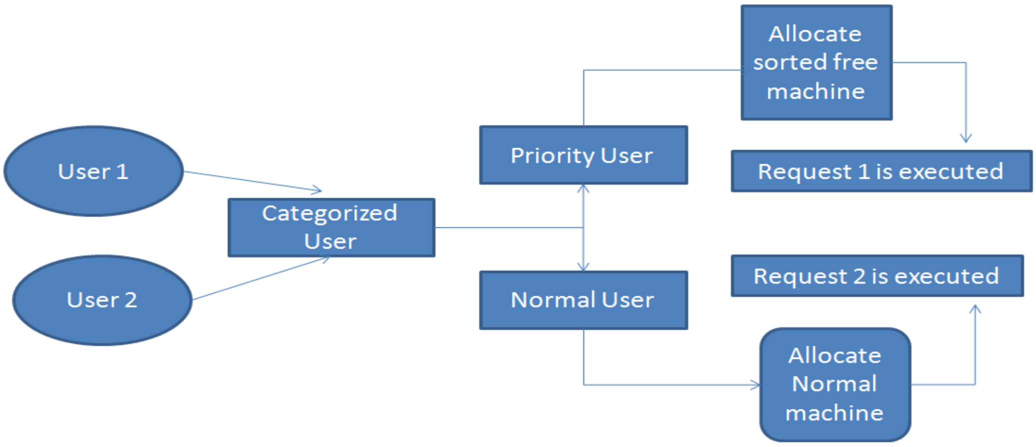

Figure 4: Proposed System

According to the proposed architecture while multiple users are requesting, Every request contain of dissimilar task. Every task different parameter are measured as time, Processor request, Importance and price. Time contains to computation time, it needed to complete the task, and Processor request refers to number of processors. More the number of processor sooner will be the completion of task. A cloud admin that is whether the user is old customer or new customer. Now, can price parameter which are charged by cloud admin.

Proposed Algorithm
Algorithm: to execute requests in terms of user base priority
Step 1: find the parameter of machine like cpu usage,free memory and response time
Step 2 Sorting cpu usage (Cp),Free memory(Mf),response time (Rt)
It will sort all values into the list.
Step 3: [Find the value K for machine which are heavily loaded or which are lightly loaded] Sort machine into queue Q.
Step 4 : Categorize users in two category, Priority user (Pu),Normal user (Nu)
Making queue (Q) of request of all users according to priority value A.
Step 5: [Assign request A1 to the machine K1, assign request A2 to the machine K2…….]
Step 6: Executing request of Pu(1),Pu(2)…….Nu(1),Nu(2)….
Step 7 : Print —Print the output

VII. CONCLUSION AND FUTURE WORK

As such cloud computing is wide region of research. One of the main topic of study is dynamic load balancing. So the next research will be focus on algorithm bearing in mind mainly parameters , load on the server and , current performance of server and user base priority. By bearing in mind these parameters, this algorithm outperforms the on hand load balancing schemes.
In prospect, we are going to slot in that into accessible cloud computing design for improved presentation and effective utilization of resources. We will believe parameters like types of load on particular server and cost optimization.

VIII. REFERENCES

[1] Deepika,D wadhwa,N kumar "Performance Analysis of Load balancing Algorithm In Distributed System " *Advance in Electronic and Electric engineering ISSN 2231-1297,volume 4*,Number 1(2014), pp.59-66,Research India Publication

[2] Sreenivas V, Prathap M,M Kemel " Load Balancing Techniques: Major challenge in Cloud computing " Electronics and Communication System (ICECS),IEEE nternational Conference, pp.1-6,2014

[3] N Shahpure,Dr.Jayrekha P "Load balancing with optimal Cost Scheduling Algorithm " *Computation of Power,Energy,Information and communication (ICCPEIC)*, IEEE International Conference, pp. 24-31,2014

[4] J Bhatia,T Patel,H Trivedi,V Majumdar "HTV Dynamic Load Balancing Algorithm for Virtual Machine Instances in Cloud " *Cloud and Services Computing(ISCOS),2012 IEEE International Symposium* pp. 15-20,December 2012.

[5] JianzheTai,JueminZhang,JunLi,WaleedMeleis and NingfangMi "A R A: Adaptive Resource Allocation for Cloud Computing Environments under Bursty Workloads" 978-1-4673-0012-4/11 ©2011 IEEE.

[6] L. Cherkasova, D. Gupta, and A. Vahdat, "When virtual is harder than real: Resource allocation challenges in virtual machine based on environments," Technical Report HPL-2007-25, February 2007.

[7] Rewinin H E, Lewis T G, Ali H H, "Task Scheduling in parallel and Distributed System Englewood Cliffs," New Jersey: Prentice Hall,1994, pp. 401-403.

[8] Wu M, Gajski D, Hypertool, "A programming aid for message passing system," IEEE Trans Parallel DistribSyst, 1990, pp. 330-343.

[9] Hwang J J, Chow Y C, Anger F D, "Scheduling precedence graphics in systems with inter-processor communication times," SIAM J Comput, 1989, pp. 244-257.

[10] Rewinin H E, Lewis T G, "Scheduling parallel programs onto arbitrary target machines," J Parallel DistribComput, 1990, pp. 138-53.

[11] Sih G C, Lee E A, “A compile-time scheduling heuristic forInterconnection-constraint heterogeneous processor architectures” IEEE Trans Parallel DistribSyst, 1993, pp. 175-187.

[12] https://devcentral.f5.com/weblogs/dmacvittie/archive/2009/03/31/introtoload-balancing-for-developers-ndash-the-algorithms.aspx

[13] KrimitShukla, Harshal Trivedi, Parth Shah “Architecture for Securing Virtual Instance in Cloud” (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 3 (3), 2012, 4279– 4282.

[14] Jitendra Bhatia, Ankit Patel, ZunnunNarmawala, “Review on variants of network coding in wireless ad-hoc networks” IEEE, Engineering (NUiCONE), 2011 Nirma University International Conference on8-10 Dec. 2011, 1 - 6

[15] http://en.wikipedia.org/wiki/Load_balancing_%28computing%29

[16] http://searchwindowsserver.techtarget.com/tip/Network-Load-Balancing-cluster- modes

[17] http://docs.openstack.org/juno/install-guide/install/apt/content/

[18] http://searchdl.org/public/conference/2014/ITC/92.pdf

[19] http://people.cse.nitc.ac.in/sites/default/files/aviralnigam/files/report.pdf