VM Level Load Balancing in Cloud Environment

Mr. M. Ajit
M. Tech. Student, Dept. of CSE,
RIT, Sakharale, Islampur.
415414, MS, India.
ajitinfo022@gmail.com

Ms. G. Vidya
M. Tech. Student, Dept. of CSE,
RIT, Sakharale, Islampur.
415414, MS, India.
vidyaa022@gmail.com

***Abstract*—As Cloud Computing is spreading globally and number of users demanding more cloud services and better results are growing rapidly, cloud load balancing become a very interesting and important research area. Generally, cloud is based on powerful datacenters that handle large number of users, so it must be featured with load balancer to achieve reliability which depends on the way it handles the load. Cloud load balancing helps to enhance the overall cloud performance. Many algorithms were suggested for assigning the users requests to Cloud resources to provide services efficiently. This paper presents the analysis of three contemporary algorithms in cloud analyst tool to resolve the issue of cloud load balancing as a preparation phase for new load balancing technique. A Weighted Signature based load balancing (WSLB) algorithm is proposed to minimize users response time. Further, this paper also provides the anticipated results with the implementation of the proposed algorithm.**

Keywords **- Cloud Computing, Load Balancing, Static/Dynamic LB.**

I. Introduction

In recent years, the internet can be represented as a cloud and the term "Cloud Computing" in computing research and industry today has the potential to make the new idea of 'computing as a utility' in the near future. Overview if cloud computing is covered in Fig.1

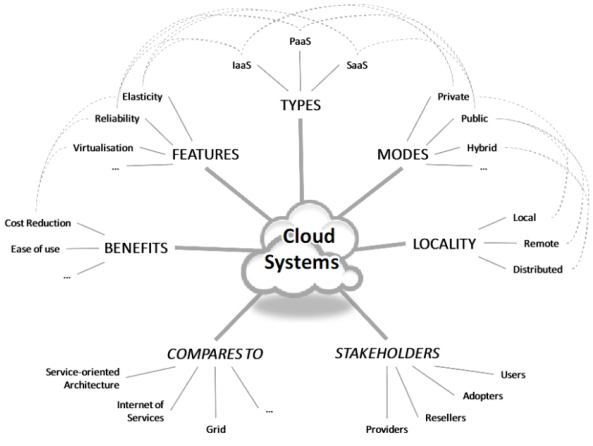

Fig. 1. View on the main aspects forming a cloud computing [13]

In IaaS, the physical resources can be split into a number of logical units called Virtual Machine (VM). User make request for resources by making an image of configuration requirement. These images get mapped on VM which is present on the server at provider side. Load balancing (LB) is done on consumer as well as on provider side[17]. On provider side,load balancing is the problem of allocating vms to servers at runtime. VM need to be reassigned so that servers do not get overload as demand changes. Just like VM load is distributed across servers, application load of client can be balanced across VMs on Consumer side. This paper addressed the load balancing issue on consumer side. All VM load balancing methods are designed to determine which VM is selected for the execution of next task units. So Resource management plays vital role in the performance of the entire cloud system and the level of user satisfaction provided by the cloud system. To provide the utmost user satisfaction, the datacenter ought to make proper resource management so as to insist the system with minimum SLA (Service Level Agreements) violation[4].

Load balancing is one of the key terms that affect the system performance dependent on the amount of work allotted to the system for a specific time period. In general, it can be described as anything that distributing computation and communication evenly among resources, or a system that divides many client requests among several servers. So there is need to manage the resources and work accordingly. The discussion on such various objectives and some LB algorithm which can be used to achieve those objectives are studied in this paper. There are several LB algorithms for the improvement and optimization of cloud performance parameters such as,

1) Throughput: The total no. of tasks that have completed execution is called throughput. A high throughput is required for better performance of the system.

2) Associated Overhead: The amount of overhead that is produced by the execution of the LB algorithm. Minimum overhead is expected for successful implementation of the algorithm.

3) Fault tolerant: It is the ability to perform correctly and uniformly even in conditions of failure at any arbitrary node in the system.

4) Migration time: The time taken in migration or transfer of a task from one machine to any other machine in the system. This time should be minimum for improving the performance of the system.

5) Response time: It is the minimum time that a distributed system executing a specific load balancing algorithm takes to respond.

6) Resource Utilization: It is the degree to which the resources of the system are utilized. A good load balancing algorithm provides maximum resource utilization.

7) Scalability: It determines the ability of the system to accomplish load balancing algorithm with a restricted number of processors or machines.

8) Performance: It represents the effectiveness of the system after performing load balancing. If all the above parameters are satisfied optimally then it will highly improve the performance of the system.

Load balancing, as the name implies, is a technique that allows workloads to be distributed across multiple resources, to make effective resource utilization and to achieve better response time by handling a condition in which some of the nodes are over loaded while some others are under loaded. The aim of load balancing is to optimize utilization and throughput while reducing the response time. As on demand self service model, load arrives randomly or dynamically in cloud computing environment which causes some VM/servers to be heavily loaded while others idle or lightly loaded which in turn leads to poor performance and make user unsatisfied.

So, if we distribute the load in proper way, it will improve system performance, throughput, response time etc. so as to meet user satisfaction.The important things to consider while developing such algorithm are: estimation of load, comparison of load, stability of different system, performance of system, interaction between the nodes, nature of work, node selection and many other ones.

Load balancing is also needed for achieving Green computing in clouds, with help of limited Energy Consumption and Reducing Carbon Emission [5]. The nature of the load balancing algorithm can be dynamic or static, although some algorithms are simple but under some conditions they work more effectively.

Static load balancing algorithms take the decision of allocating the load to VMs in system based on prior knowledge about the applications and resources of the system. The performance of the VMs is determined at the time of request arrival. Static load balancing algorithms are non-preemptive and therefore each machine has at least one task assigned for itself. Its objective is to minimize the execution time and delay and limit communication overhead.

Dynamic load balancing algorithms make any decision for load balancing based on dynamically changing state of the system. It allows for processes to move from an over utilized machine to an underutilized machine dynamically for faster execution. This means that dynamic load balancing is preemptive which helps in improving the overall performance of the system by migrating the load dynamically.

The rest of this paper is organized as follows: Section II presents a review of related work. Section III describes existing load balancing algorithms in cloud analyst tool. The proposed heuristics and result analysis is presented in Section IV. The paper is summarized and concludes in Section V.

II. RELATED WORKS

The motivation of the survey of existing load balancing techniques in cloud computing is to encourage the amateur researcher to contribute in developing more efficient load balancing algorithms. This will benefit interested researchers to carry out further work in this research area. The existing load balancing algorithms prevalent in cloud environment are,

A. Vector Dot[1]

Environment used: Datacenters with integrated server and storage virtualization.
It uses dot product to distinguish node based on the item requirement. It handles hierarchical and multidimensional resource constraints and removes overloads on server, switch and storage.

B. Carton [2]

Environment used: Unifying framework for cloud control.
It uses Load balancing to minimize the associated cost and uses Distributed Rate Limiting for fair allocation of resources. It is simple and Easy to implement and very low computation and communication overhead.

C. Compare and Balance [3]

Environment used: Intra-Cloud.
It is based on sampling process and uses adaptive live migration of virtual machines. It balances load amongst servers and reaches equilibrium fast. It assures migration of VMs from high-cost physical hosts to low cost host. It assumes that each physical host have enough memory.

D. Scheduling strategy on LB of VM resources [5]

Environment used: Cloud Computing.
It uses Genetic algorithm, historical data and current state of system to achieve best load balancing and to reduce dynamic migration.

E. LBVS [6]

Environment used: Cloud Storage.
It Uses Fair-Share Replication strategy to achieve Replica Load balancing module which in turn controls the access load balancing and uses writing balancing algorithm to control data writing load balancing.

F. Honeybee Foraging Behavior [7]

Environment used: Large scale Cloud Systems.
It achieves global load balancing through local server action.

G. Biased Random Sampling [7]

Environment used: Large scale Cloud systems.
It achieves load balancing across all system nodes using random sampling of the system domain.

H. Active Clustering [7]

Environment used: Large scale Cloud systems.
It optimizes job assignment by connecting similar services by local re-wiring.

I. ACCLB [8]

Environment used: Open Cloud Computing Federation.
It uses small-world and scale-free characteristics of complex network to achieve better load balancing.

J. OLB + LBMM [9]

Environment used: Three-level Cloud Computing Network.
It uses OLB (Opportunistic Load Balancing) to keep each node busy and uses LBMM (Load Balance Min-Min) to achieve the minimum execution time of each task.

K. Server-based LB [10]

Environment used: Distributed web servers.
It uses a protocol to limit redirection rates to avoid remote servers overloading and uses a middleware to support this protocol. It uses a heuristic to tolerate abrupt load changes.

L. Join-Idle-Queue [11]

Environment used: Cloud data centers.
It first assigns idle processors to dispatchers for the availability of the idle processors at each dispatcher and then assigns jobs to processors to reduce average queue length of jobs at each processor. Table 1 gives various objectives addressed in the literature.

TABLE I
VARIOUS OBJECTIVES FOCUSED IN LITERATURE

Objectives	**Reference Papers**
Minimization of Response Time	[7],[13],[14]
Maximization of Resource utilization	[1],[2],[3],[4],[5],[11]
Throughput	[9],[10],[15]
Performance	[2],[7],[9],[10],[11],[13],[14],[15]
Scalability	[4],[7],[9],[10],[13]
Overhead	[2],[3],[5],[14]
Fault tolerance	[7]
Migration time	[3]

III. LOAD BALANCING ALGORITHMS

After studying various load balancing algorithms, it is necessary to execute at least two or three algorithms to observe where they are improving and lacking as well. In real scenarios, executions of such algorithms are not possible in exact cloud computing environment. As a consequence, simulating of these algorithms might be supportive in order to accomplish this research. So research work is done on cloud analyst simulation tool to resolve the cloud load balancing issue as a preparation phase for new load balancing technique. Load balancing algorithms present in cloud analyst are,

A. Round Robin:

As name implies, it is simplest load balancing algorithm uses the time slicing mechanism. It works in the round trip where a time is divided into slices and is allotted to each node. Each node has to wait for their turn to perform their task. This algorithm has less complexity as compared to the other two algorithms. Open source simulation software knows as cloud analyst uses this algorithm as default algorithm in the simulation. This algorithm has less complexity as compared to the other two algorithms. This algorithm simply assigns the jobs in round robin fashion without considering the load on different machines. Though the algorithm is very simple, there is an additional load on the scheduler to decide the size of time slice and it has longer average waiting time, higher context switches higher turnaround time and low throughput.

B. Equally Spread Current Execution Load(ESCE):

As name implies, in this algorithm, load balancer makes an effort equally spreading the execution load on different VMs. Load balancer maintains an index table of VMs along with the number of requests currently allocated to the VM. If there is request comes from the data centre for execution, load balancer search the index table for least loaded VM. If more than one VM is found, first identified VM is selected and allotted for request execution. The load balancer updates the index table by increasing the allocation count of identified VM. When VM finishes the execution of allotted request,load balancer again update the index table by decreasing the allocation count for identified VM by one. So, there is an additional computation load balancer to search the table again and again.

C. Throttled Load Balancing:

In this algorithm the load balancer maintains an index table of VMs along with their states (Available or Busy). Initially, all VMs state is Available. If there is request comes from the data centre for execution, load balancer search the index table from top until the first available VM is found or the index table is searched fully. If the VM is found, the request is allotted to that VM and load balancer update the index table accordingly by making VMs state Busy. When the VM completes the allotted request, load balancer updates the index table accordingly by making VMs state Available. These algorithms are being experimentally performed using the cloud analyst tool which helps in testing the outputs with respects to the virtual machine. If no virtual machines are available to execute jobs immediately then the client request get queued for execution till VM becomes available.

IV. PROPOSED WORK AND RESULT ANALYSIS

Cloud analyst is used to test the performance of these three algorithms and compare them with proposed one with respect to the response time. Cloud analyst simulation tool is based on cloudsim library written in java and provides a GUI interface to configure various parameters to perform the experimental work. Fig. 2 shows cloudsim components considered in the environment of cloud analyst simulation tool used for experimental work.

This research work considers Datacenter, VM, host and Cloudlet components from CloudSim for implementation of a proposed algorithm. Datacenter component handles service requests. VM consist of application elements which are connected with these requests, so Datacenters host should allocate VM requested by user. Cloud Analyst can evaluate

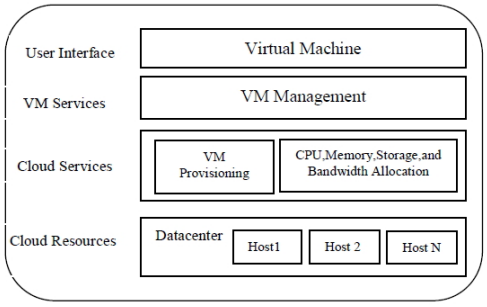

Fig. 2. Environment for experimental work.

any algorithm or application deploying in the cloud [16]. VM life cycle starts from provisioning of a host to a VM, VM creation, VM destruction, and VM migration [16].

A brief description of these components and the working relationship between them is presented in the following:

1) Datacenter: Each Datacenter has a number of hosts with homogeneous or heterogeneous configurations (memory, cores,capacity, and storage). It also creates the bandwidth, memory,and storage devices allocation.

2) Virtual Machine (VM): VM characteristics comprise of memory, processor, storage, and VM scheduling policy.Multiple VM can run on single hosts simultaneously and maintain processor sharing policy. The research work considers the demands for VMs with homogeneous configuration.

3) Host: Hosts are present in the datacenter with homogeneous or heterogeneous configuration. This experiment considers VM need to handle a number of cores to be processed and host should have resource allocation policy to distribute them in these VMs. So host can arrange sufficient memory and bandwidth to the process elements to execute them inside VM. Host handles creation and destruction of VMs.

4) Cloudlet: Cloudlet is an application task which is responsible to deliver the data in the cloud service model which get executed on VM mapped on host.

5) VM Provisioner: It is used to allocate host for demanded VM. In cloud analyst VMs are randomly mapped on VM, and requests are executed on VM selected by load balancer as per load balancing policy.

In this paper, we have proposed the VM load balancing algorithm at the VM level where, individual task is assigned to VM mapped on host with different computing power. So the load assignment factor is determined for each host based on its configuration to achieve optimization goal such as minimization overall response time. The tasks/requests are assigned or allocated to the VM mapped on host with highest load assignment factor and then to the lowest and so on.

This research work done into three parts: calculation and assignment of load assignment factor to hosts, provisioning and balancing. Load assignment factor is used to identify the host with high configuration that gives faster response than the other and is assigned to host based on the value calculated as in [15]

$$Capacity_{host} = \sqrt{CPU^2 + Memory^2 + Storage^2} \quad (1)$$

Host with high capacity value get highest load assignment factor. Provisioner allocate host to demanded VM according to the free processing elements. Once demanded VMs are get allotted on host, load get distributed on these VM according to load balancing policy. When request arrives at datacenter, datacenter call load balancer for VM to execute request. Load balancer searches list of VMs mapped on host having high load assignment factor. Once found, it select the VM which is found available first and send to the datacenter for allocation and update the status of that VM as busy.

Following are the steps performed in our algorithm,

Step1: Creation of host within datacenter with homogeneous or heterogeneous configuration.

Step2: Calculation and assignment of load assignment factor to created hosts in a datacenter. Homogeneous hosts has same load assignment factor whereas heterogeneous hosts has varying load assignment factor according to configuration.

Step3: Provisioning of VMs. In this step, hosts are assigned to VMs demanded by user for their execution. Here the host with highest load assignment factor has highest priority for mapping VM.

Step4: Load balancer keeps the track of all VMs along with their allotted host and status (Available/Busy). At start status of all VMs is Available.

Step5: When request arrives from datacenter controller to allocate VM for execution, it search all the VMs and identifies powerful VM in the sense it is mapped on host with highest load assignment factor which is available to take the request immediately. If no VM is available on that host, lowest one is searched for available VM. Selected VM is then returned to datacenter controller.

Step6: Datacenter controller allocates the request to selected VM return by load balancer and notifies load balancer about this new allocation.

Step7: Load balancer updates the status of that VM as busy.

Step8: When request execution completed, datacenter controller de-allocates that VM and notifies load balancer about this de-allocation.

Step9: Load balancer updates the status of that VM as available.

Step10: continue from step5 until all requests get processed.

We execute these algorithm steps by assuming there is one data center having 3 hosts with configuration given in Table II, and 40 VMs with 1024MB memory and 1000 bandwidth demanded by setting configuration given in Table II,III, IV.

TABLE II
HOST CONFIGURATION

Host ID	Memory(MB)	Storage(MB)	No. of Processors
0	204800	100000000	4
1	102400	50000000	2
2	51200	25000000	1

TABLE III
ADVANCED CONFIGURATION

User grouping factor	1000
Request grouping factor	100
Executable instruction length	250

TABLE IV
USER BASE CONFIGURATION

User Base	Request per user per hour	Data Size per request
UB1	12	100
UB2	12	10000
UB3	12	100000
UB4	12	1000
UB5	12	10000

Algorithms give average response time(RT)and Data Center Request Servicing Time(DC-RST) with these parameters setting in the same region as well as in the different region as shown in Table V and Table VI respectively.

TABLE V
AVERAGE RESPONSE TIME OBTAINED

Parameters	Load Balancing Algorithms			
	RR	**ESCE**	**Throttled**	**WSLB**
Avg. RT(ms)	315.03	314.06	310.51	**269.06**
Avg. DC-RST(ms)	9.88	9.05	5.64	**10.59**

TABLE VI
AVERAGE RESPONSE TIME OBTAINED

Parameters	Load Balancing Algorithms			
	RR	**ESCE**	**Throttled**	**WSLB**
Avg. RT(ms)	353.05	352.79	349.57	**302.81**
Avg. DC-RST(ms)	10.1	9.06	5.68	**5.68**

From this, it is seen that our proposed method gives better response time than the contemporary algorithms. It is possible because virtual machine is placed on heterogeneous host based on its processing performance and host with high configuration is faster than the others. It is also observed that DC servicing time is more than other three. Because, in contemporary algorithms, there is searching of VM list only,but our algorithm searches the host list to find host with maximum load assignment factor and then searches the all VMs placed on that host for available one. There is extra overhead for searching host than the others. So DC requires more serving time than other. As compared to average response time reduction, it is not as much considerable.

V. CONCLUSION AND FUTURE WORK

This paper proposed a new VM load balancing algorithm: Weighted Signature based Load Balancing (WSLB) and implemented it using cloud analyst tool with help of CloudSim library (an abstract cloud computing environment) using java language. Our algorithm find the load assignment factor for each of the host in a datacenter and map the VMs according to that factor. Load balancer sends virtual machine id which is available on highest configuration host having maximum load assignment factor then lowest one and so on. According to the experimental results, it conclude that if we select a virtual machine mapped on powerful host, then it affects the overall performance of the cloud Environment and decrease the average response time. Proposed method work out on homogeneous VMs mapped on hosts. Mapping of heterogeneous VMs based on dynamic workload on heterogeneous hosts will be considered in future for further improvement.

REFERENCES

[1] A.Singh, M.Korupolu, D.Mohapatra, "Server-storage virtualization:integration and load balancing in data centers," *Proceedings of the ACM/IEEE conference on Supercomputing (SC)*, pp. 978-1-4244-2835-9/08, 2008.

[2] R.Stanojevic, R.Shorten, "Load balancing vs.distributed rate limiting: a unifying framework for cloud control," *Proceedings of IEEE ICC,Dresden, Germany*, pp. 1-6, 2009.

[3] Y. Zhao, W.Huang, "Adaptive Distributed Load Balancing Algorithm based on Live Migration of Virtual Machines in Cloud," *Proceedings of 5th IEEE International Joint Conference on INC, IMS and IDC,Seoul, Republic of Korea*, pp. 170-175, 2009.

[4] L.Deboosere, V.Bert, S.Pieter, D.T.Filip,D.Bart and D.Piet, "Efficient resource management for virtual desktop cloud computing," *Springer*, 2012.

[5] J.Hu,J.Gu, G.Sun, T.Zhao, "A Scheduling Strategy on Load Balancing of Virtual Machine Resources in Cloud Computing Environment," *Third International Symposium on Parallel Architectures,Algorithms and Programming (PAAP)*, pp. 89-96, 2010.

[6] H.Liu, S.Liu, X.Meng, C.Yang, Y.Zhang, "LBVS: A Load Balancing Strategy for Virtual Storage," *International Conference on Service Sciences (ICSS), IEEE*, pp. 257-262, 2010.

[7] M.Randles, D.Lamb, A.Taleb-Bendiab, "A Comparative Study into Distributed Load Balancing Algorithms for Cloud Computing," *Proceedings of 24th IEEE International Conference on Advanced Information Networking and Applications Workshops, Perth, Australia*, pp. 551-556, 2010.

[8] Z.Zhang, X.Zhang, "A Load Balancing Mechanism Based on Ant Colony and Complex Network Theory in Open Cloud Computing Federation," *Proceedings of 2nd International Conference on Industrial Mechatronics and Automation (ICIMA),Wuhan, China*, pp. 240-243, 2010.

[9] S.Wang, K.Yan, W.Liao, S.Wang, "Towards a Load Balancing in a Three-level Cloud Computing Network," *Proceedings of the 3rd IEEE,International Conference on Computer Science and Information Technology (ICCSIT),Chengdu, China*, pp. 108-113, 2010.

[10] A.M.Nakai, E.Madeira, L.E.Buzato, "Load Balancing for Internet Distributed Services Using Limited Redirection Rates," *5th IEEE,Latin-American Symposium on Dependable Computing (LADC)*, pp. 156-165, 2011.

[11] Y.Lua, Q.Xiea,G.Kliotb, A.Gellerb, J.R.Larusb,A.Greenber, "Join-Idle-Queue: A novel load balancing algorithm for dynamically scalable web services," *An international Journal on Performance evaluation, In Press, Accepted Manuscript*, 2011.

[12] X.Liu, L.Pan, Wang, Chong-Jun, Xie, Jun-Yuan, "A Lock-Free Solution for Load Balancing in Multi-Core Environment," *3rd IEEE International Workshop on Intelligent Systems and Applications (ISA)*, pp. 1-4, 2011.

[13] S.Lutz, "The Future of Cloud Computing,", 2010.

[14] T.Wei-Tek, S.Xin, B.Janaka, "Service-Oriented Cloud Computing Architecture," *Computer society*, 2010.

[15] Lee Rich, and Jeng Bingchiang, "Load-Balancing Tactics in Cloud ," *System Technology Group of IBM Corporation, Taiwan,International Conference*, 2011.

[16] Bhathiya Wickremasinghe, "Cloud Analyst: A Cloud Sim-based Visual Modeler for Analyzing Cloud Computing Environments and Applications ,", 2010.

[17] Ilyas Iyoob,ChainOpt,Gravitant Emrah,Zarifoglu, and Gravitant A.B. Dieker, "Cloud Computing Operations Research" ,*Georgia Institute of Technology*,2011.