A novel approach for load balancing in cloud data center

Conference Paper · February 2014

DOI: 10.1109/IAdCC.2014.6779427

Mala Kalra
National Institute of Technical Teachers Training and Research,Chandigarh,India

A Novel Approach for Load Balancing in Cloud Data Center

Gulshan Soni[1],
[1]*M.E. Student, Department of Computer Science,*
NITTTR, Chandigarh, India
[1]gsoni260@gmail.com,

Mala Kalra[2]
[2] *Assistant Professor, Department of Computer Science,*
NITTTR, Chandigarh, India
[2] malakalra2004@gmail.com

Abstract- **In a large-scale cloud computing environment the cloud data centers and end users are geographically distributed across the globe. The biggest challenge for cloud data centers is how to handle and service the millions of requests that are arriving very frequently from end users efficiently and correctly. In cloud computing, load balancing is required to distribute the dynamic workload evenly across all the nodes. Load balancing helps to achieve a high user satisfaction and resource utilization ratio by ensuring an efficient and fair allocation of every computing resource. Proper load balancing aids in minimizing resource consumption, implementing fail-over, enabling scalability, avoiding bottlenecks and over-provisioning etc. In this paper, we propose "Central Load Balancer" a load balancing algorithm to balance the load among virtual machines in cloud data center. Results show that our algorithm can achieve better load balancing in a large-scale cloud computing environment as compared to previous load balancing algorithms.**

Keywords— ***Load balancing, Cloud Data Center, Live Virtual Machine Migration, Virtualization, CloudAnalyst***

I. INTRODUCTION

The cloud means the applications and services that are offered from data center to all over the world. These applications and services are offered over the internet. The services provide by cloud computing are infrastructure as a service (IaaS), platform as a service (PaaS) and software as a service (SaaS) that are made available as pay-as-you-go model to clients. Cloud Computing Deployment Model refers to the location and management of the infrastructure cloud services. The Deployment Model of cloud computing are Private Cloud, Community cloud, Public cloud and Hybrid cloud. Cloud Computing contain some essential characteristics that are rapid elasticity, on-demand self-service, resource pooling, broad network access, and measured service. Cloud computing ... Virtualization is a ...l servers that run on ...ualization allows for ...s to create a larger ...educing the overall ...re.

Currently cloud computing is becoming popular among users and corporate world but despite of growing uses of cloud technology, many crucial problems still need to be solved for the realization of cloud computing. Load balancing is one of these problems, it plays a very important role in the realization of Cloud Computing. Load balancing means the ability to distribute the load over a number of separate systems therefore the overall performance of processing the incoming requests increased. There are four major resources processor (CPU), memory (RAM), network and storage (Disk). In traditional computing environments, researchers [16, 17, and 18] have proposed various static, dynamic and mixed load balancing policies. Static load balancing algorithm assign load to machines according to their processing capability but do not consider dynamic changes of these attributes at run-time. Commonly used static algorithms are Round Robin (RR) & Weighted Round Robin (WRR). Dynamic load balancing algorithm collects information and run times conditions of machines and according to gathered characteristics assign and dynamically reassign the load among machines. Least connection (LC) and weighted least connection (WLC) are dynamic load balancing algorithms commonly used.

In the cloud computing environment, load balancing is required to achieve short response time and high system throughput. For cloud environment various load balancing approaches have been proposed such as Honeybee-based load balancing technique [3], Active Clustering [3], Random sampling [3], Active Monitoring Load Balancer [4], Throttled Load Balancer [4], WCAP [6], JIQ [7], CLBVM [13] etc.

The rest of this paper is organized as follows: Section II gives a review of the related work which realizes load balancing importance in cloud computing. Section III introduces the proposed approach for load balancing in cloud. Section IV describes the experimental setup for implementation of the proposed algorithm. Section V analyses the performance of the mechanism. Finally conclusion of the work is discussed in Section VI along with the envisaged future work.

II. MOTIVATION AND RELATED WORK

Various load balancing algorithm have been proposed for cloud computing to provide efficient distribution of load among available machines. A number of techniques proposed for load balancing are based on live virtual machine migration. **Ma et al.** [15] proposed a new model for distributed load balancing allocation of virtual machine in cloud data center using the TOPSIS method which is one of the most efficient Multi Criteria Decision Making (MCDM) technique. This method can find the most suitable physical machine in the data center for the migrated VMs. MCDM technique try to avoid the live virtual machine migration. **Zhao et al.** [11], proposed a distributed load balancing algorithm COMPARE_AND_BALANCE based on sampling to reach an equilibrium solution. They designed and implemented a simple model which decreases the migration time of virtual machines by shared storage and fulfills the zero-downtime relocation. Live virtual machine migration has mainly two performance issues:

1) Total migration time: It is total time taken to migrate virtual machines from its host machine to the target machine.

2) Down time: Down time is duration of time at which services are not available to the users.

Randles et al. [3], introduced three load balancing algorithm in large-scale complex systems are an extended honeybee foraging algorithm, a biased random sampling on a random walk procedure and Active Clustering. The inspiration from the Honeybee algorithm consisting servers s1,…..sn are arranged into M virtual servers VS0,….VSM-1 with service queues Q1,…,QM-1 respectively. The reward for a server SiєSj serving a request from Qi is cj. Biased Random Sampling approach instead of monitoring the nodes and their available resources through a static network, a dynamic network system is created that provides a measure of instant load distribution status, and gives dynamics for job allocation and resource update. Active Clustering considered, as a self-aggregation algorithm, to rewire the network. Application of this procedure is intended to group like-service instances together as many load balancing algorithms only work well in cases where the nodes are aware of their like nodes and can easily delegate workload to them.

g mechanism based
eory in open cloud
spects of the related
ed to realize load

ontent aware load
client aware policy
oach of client aware
ation policy.

Lua et al. [7], addressed Join-Idle-Queue (JIQ) algorithms for distributed load balancing in large systems. The JIQ algorithm incurs no communication overhead between the dispatchers and processors at job arrivals.

Liu et al. [8], define 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.

Liu et al. [9], suggested a load balancing virtual storage strategy (LBVS) that provides a large scale net data storage model and Storage as a Service model based on Cloud Storage. Y. Fang et al. [10], proposed a two-level task scheduling mechanism based on load balancing to meet dynamic requirements of users and obtain a high resource utilization.

Wang et al. [12], proposed scheduling algorithm which combines OLB (Opportunistic Load Balancing) and LBMM (Load Balance Min-Min) scheduling algorithms that can utilize better executing efficiency and maintain the load balancing of system.

Bhadani et al. [13], recommended Central Load Balancing Policy for Virtual Machines (CLBVM) to balance the load evenly in a distributed virtual machine/cloud computing environment.

Zhang el al. [14], introduced an approach (Statistic based Load Balance, SLB) that makes use of the statistical prediction and available resource evaluation mechanism to make online resource allocation decisions.

Bhathiya et al. [4] proposed two virtual machine load balancing algorithms, which have been used for load balancing in cloud data center. First algorithm is Active Monitoring Load Balancer, which distributes the load equally to available virtual machines in the way that each virtual machine consist equal number of tasks. Second algorithm is Throttled Load Balancer, which ensures only a pre-defined number of task/request are allocated to a single VM at any given time. If requests are present more than pre-defined number of VM's at a data center, than some of the requests will have to be queued until the next VM becomes available. Load balancing algorithms Active Monitoring Load Balancer [4] and Throttled Load Balancer [4] worked properly when all the virtual machines of data center had similar hardware configurations. The major problem occurs when the hardware configuration of virtual machines is different and it creates the under load and over load situations in virtual machines. The key challenge is to develop a load balancing algorithm, which will achieve the better load balancing among virtual machines that had different hardware configurations in cloud data center.

III. PROPOSED APPROACH FOR LOAD BALANCING

The proposed load balancing algorithm "Central Load Balancer" will balance the load among virtual machines having different hardware configurations and will distribute the load based on hardware configuration and states of virtual machines in data center. The proposed technique will be able to perform quick and reliable load balancing in cloud computing environment through utilization of all virtual machines according to their computing capacities.

In the proposed technique, every request from user bases arrive at Data Center Controller. Data Center Controller queries the Central Load Balancer for allocation of requests. Central Load Balancer maintain a table that consist of id, states and priority of virtual machines. Central Load balancer parses the table and find out highest priority virtual machine, then check its states and if its states available then return that virtual machine id (VMid) to Data Center Controller. If the states of virtual machine is Busy then it chooses next less high priority virtual machine. Finally Data Center Controller assigns the request to that VMid that is provided by Central Load Balancer (CLB).

The Central Load Balancer (CLB) is connected to all users and virtual machines present in cloud data center through Data center Controller as shown in Figure 3.1. The Central Load Balancer calculates the priorities of virtual machines based on their CPU speed (MIPS) and memory.

Virtual Machine Priority calculation

The priority of each virtual machine is calculated based on its CPU speed (MIPS) and memory. It is calculated as following:

$$\boxed{\mathbf{Pr(i) = t^* \, Tc(i) \; + s^* \, Tm(i)}}$$

. Where $(1 \leq i \leq n)$ and $t + s = 1$

Notations are following:

Pr= Priority of each virtual machine node,
Tc = CPU speed (MIPS),
Tm = memory resource,
t = the CPU weight,
s = the weight of memory

s demonstrate in the

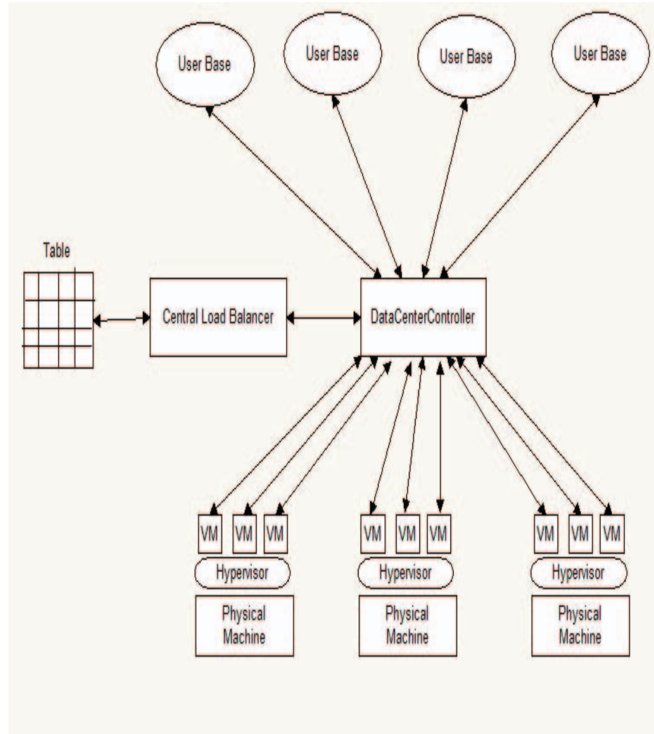

Figure 3.1. Working of Central Load Balancer

The proposed algorithm is shown in Figure 3.1 .The description of proposed algorithm is following

1. Central Load Balancer maintains a table that contains virtual machine id (VMid), states (BUSY/AVAILABLE) and priority of VMs. Initially, all Virtual Machines are in available state.
2. Data Center Controller receives a new request.
3. Data Center Controller queries the Central Load Balancer for next allocation.
4. Central Load Balancer parses the table from top to find the highest priority virtual machine and the state of that virtual machine's is available.
   If found:
   a) The Central Load Balancer returns the VMid to the Data Center Controller.
   b) The Data Center Controller sends the request to the VM identified by that VMid.
   c) Data Center Controller notifies the Central Load Balancer of new allocation.
   d) Central Load Balancer updates the table accordingly.

   If not found:

   e) The Central Load Balancer returns -1.
   f) The Data Center Controller queues the request.

5. When the VM finishes processing of requests, and the Data Center Controller receives the response cloudlet, it notifies the Central Load Balancer of the VM de-allocation.
6. The Data Center Controller checks if there are any waiting requests in the queue. If there are, it continues from step 3.
7. Continue from step 2.

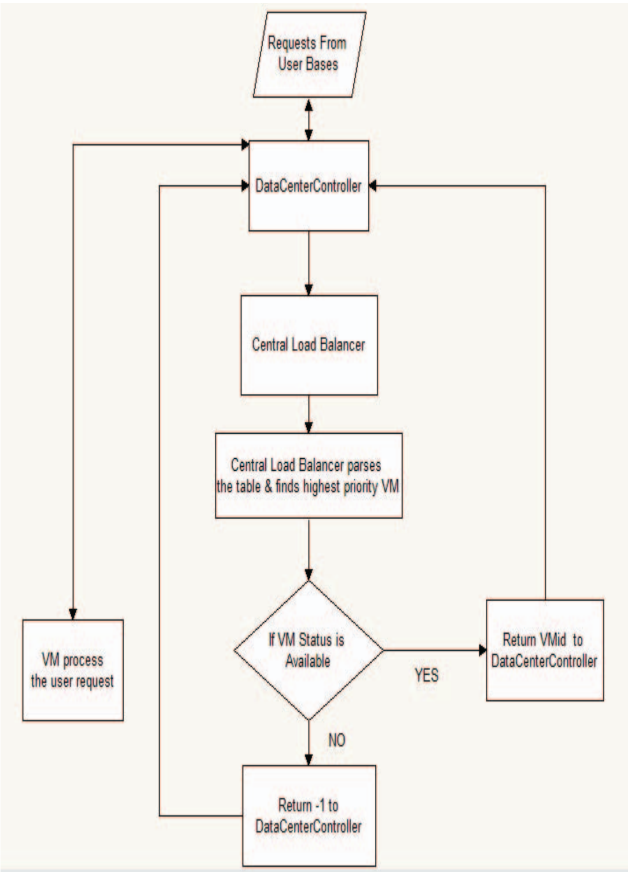

Figure 3.2. Proposed algorithm

IV. EXPERIMENTAL SETUP

The proposed algorithm is implemented and integrated in CloudAnalyst tool [4].The CloudAnalyst is a CloudSim-based tool for modelling and analysis of large cloud computing enviroment.

that is 1/10th of the

6 region with the

Table 4.1 Define 6 user bases representing the 6 region

Name	Region	Req. per User per Hr	Data Size per Req. (bytes)	Peak Hrs strart (GMT)	Peak Hrs End (GMT)	Avg Peak Users	Avg off-Peak Users
UB1	0	60	100	13	15	400000	40000
UB2	1	60	100	15	17	100000	10000
UB3	2	60	100	20	22	300000	30000
UB4	3	60	100	1	3	150000	14000
UB5	4	60	100	21	23	50000	5000
UB6	5	60	100	9	11	80000	8000

Application Deployment Configuration
Service Broker Policy: Closest Data center

Other parameter used are given in the table below

Table 4.2 Parameter Used

Parameter	Value Used
VM Image Size	10000
VM Memory	1024 Mb
VM Bandwidth	1000
Data Center – Architecture	X86
Data Center – OS	Linux
Data Center – VMM	Xen
Data Center – Number of Machines	20
Data Center – Memory per Machine	2048 Mb
Data Center – Storage per machine	100000 Mb
Data Center – Available BW per Machine	10000
Data Center – Number of processors per machine	4
Data Center – Processor speed	100 MIPS
Data Center – VM Policy	Time Shared
User Grouping Factor	1000
Request Grouping Factor	100
Executable Instruction Length	250

Latency Matrix values (in millisecond) and Bandwidth Matrix values (in Mbps) table is same as provided in CloudAnalyst [4].

V. PERFORMANCE EVALUATION

To test the efficiency of our VM load balancing algorithm, we are considering two test cases. In first test case load is kept constant and the number of virtual machines are increased and second case number of virtual machines are kept constant (10) and the load is increased constantly through alter data size per request. We carried out our experiment based on all the parameters and user base table mentioned in section IV and like with most real-world web application let us assume initially the application is deployed in a single location, in Region 0 (North America).

Case 1: Load is kept constant and the number of virtual machines will be varied from 5 to 40.

We have observed that the response time of our proposed algorithm "Central Load Balancer" is less as compared to other three load balancing policies, which are proposed by Bhathiya et al. [4]. The Comparison of four algorithms when system load is stable as below:

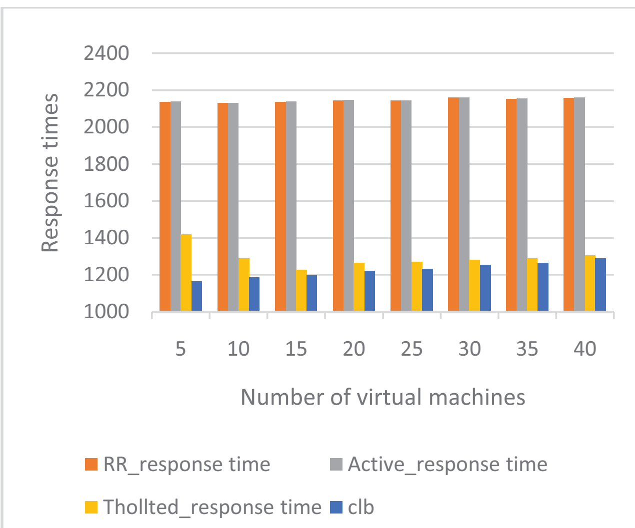

Figure 5.1 Comparison of four algorithms when system load is stable

Case 2: Number of virtual machines are kept constant (10) and the load is increased through alter data size per request:

...e of our proposed
...ss as compared to
...h are proposed by
...orithms when no. of

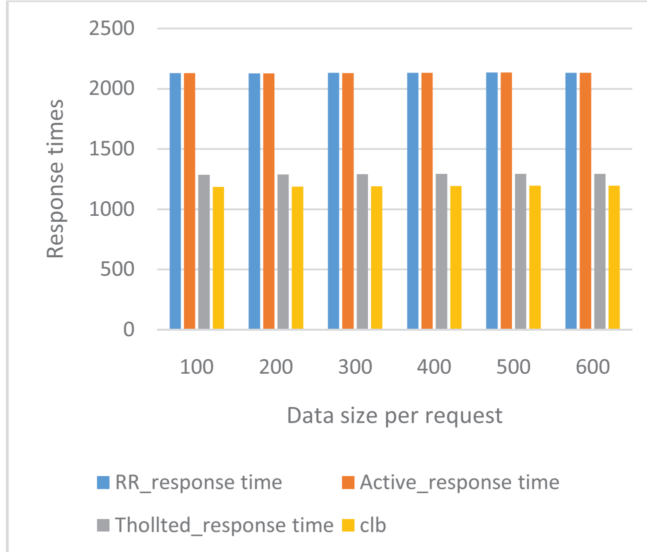

Figure 5.2 Comparison of four algorithms when no. of VMs is stable

VI. CONCLUSION AND FUTURE WORK

In proposed Central Load Balancer (CLB) technique, we tried to avoid the situation of over loading and under loading of virtual machines. The Central Load Balancer (CLB) manages load distribution among various virtual machines and assigns load corresponding to their priority and states. In this way this technique efficiently shares the load of user requests among various virtual machines.

The future work is to develop a load balancing algorithm that will allocate the load to virtual machines according to their current resource utilization such as current utilization of processor and memory so the load distribution will be more dynamic and robust.

REFERENCES

[1] M. Nelson, B. Lim, and G. Hutchins, "Fast transparent migration for virtual machines", in Proceedings of the annual conference on USENIX Annual Technical Conference , pp. 25-25, April 2005.

[2] M. Hines and K. Gopalan, "Post-copy based live virtual machine migration using adaptive pre-paging and dynamic self-ballooning", in Proceedings of the ACM/Usenix International Conference on Virtual Execution Environments (VEE'09), pp.51-60, March 2009.

[3] M. Randles, D. Lamb, and A. Bendiab, "A comparative Study into distributed load balancing algorithms for cloud computing", IEEE 24th International Conference on Advanced Information Networking and Applications Workshops (WAINA), pp.551-556, April 2010.

[4] W. Bhathiya, "CloudAnalyst a CloudSim-based tool for modelling and analysis of large scale cloud computing environments", MEDC Project, Cloud Computing and Distributed Systems Laboratory, University of Melbourne, Australia, pp. 1-44, June 2009.

[5] Z. Zhang and Xu. Zhang "A Load balancing mechanism based on ant colony and complex network theory in open cloud computing federation", 2nd International Conference on Industrial Mechatronics and Automation (ICIMA), Wuhan, China, vol. 2, pp.240-243, May 2010.

[6] H. Mehta, P. Kanungo, and M. Chandwani, “Decentralized content aware load balancing algorithm for distributed computing environments”, Proceedings of the International Conference Workshop on Emerging Trends in Technology (ICWET), pp. 370-375, February 2011.

[7] Y. Lua, Q. Xiea, G. Kliotb, A. Gellerb, J. R. Larusb, and A. Greenber, “Join-Idle-Queue: A novel load balancing algorithm for dynamically scalable web services”, An international Journal on Performance Evaluation, vol. 68, pp.1056-1071, November 2011.

[8] X. Liu, Pan, C. Wang, and J. Xie, “A lock-free solution for load balancing in multi-core environment”, 3rd IEEE International Workshop on Intelligent Systems and Applications (ISA), pp. 1-4, May 2011.

[9] H. Liu, S. Liu, X. Meng, C. Yang, and Y. Zhang, “LBVS: A load balancing strategy for virtual storage”, International Conference on Service Sciences (ICSS), pp. 257-262, May 2010.

[10] Y. Fang, F. Wang, and J. Ge, “A task scheduling algorithm based on load balancing in cloud computing”, International Conference Web Information Systems and Mining (WISM 2010), Vol. 6318, pp. 271-277, October 2010.

[11] Y. Zhao, and 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, August 2009.

[12] S. Wang, K.Yan , W. Liao and S. Wang , “Towards a load balancing in a three-level cloud computing network”, 3rd IEEE International Conference on Computer Science and Information Technology (ICCSIT), vol. 1, pp. 108 - 113, July 2010.

[13] A. Bhadani, and S. Chaudhary, “Performance evaluation of web servers using central load balancing policy over virtual machines on cloud”, Proceedings of the Third Annual ACM Bangalore Conference (COMPUTE’10), Article No. 16, January 2010.

[14] Z. Zhang, H. Wang, L. Xiao and L. Ruan, “A statistical based resource allocation scheme in cloud”, IEEE International Conference on Cloud and Service Computing (CSC), pp. 263-273, December 2011.

[15] F. M a, F. Liu and Z. Liu, “Distributed load balancing allocation of virtual machine in cloud data center”, IEEE 3rd International conference on Software Engineering and Service Science (ICSESS), pp.20-23, June 2012.

[16] C. Sundaram, Y. Narahari, “Analysis of dynamic load balancing strategies using a combination of stochastic Petri nets and queueing networks”, Proceedings of the 14th International Conference Chicago, Illinois, USA, pp. 397-414, June 1993.

[17] S. Sharma, S. Singh, and M. Sharma, “Performance analysis of Load Balancing Algorithms”, World Academy of Science, Engineering and Technology (WASET), vol. 14, pp. 248-251, February 2008.

[18] Y. Teo and R. Ayani, “Comparison of load balancing strategies on cluster-based web servers, Simulation”, the journal of the Society for Modeling and Simulation, vol. 77, pp. 185-189, November- December 2001.