# ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of any task would be, but impossible without the mention of the people who made it possible, whose constant guidance and encouragement crowned our efforts with success.

I thank the management, **Dr. Mohan Manghnani**, Chairman of NEW HORIZON EDUCTIONAL INSTITUTIONS for providing necessary infrastructure and creating good environment.

I also record here the constant encouragement and facilities extended to us by **Dr. Manjunatha**, Principal, NHCE and **Dr. Prashanth.C.S.R,** Head of the Department of Computer Science and Engineering. I extend sincere gratitude to them.

I sincerely acknowledge the encouragement, timely help and guidance to me by our beloved guide **Ms.SURIYA REFAI BEGUM**, to complete the seminar within the stipulated time successfully.

Finally a note of thanks to the teaching and non-teaching staff of Computer Science and Engineering Department for their cooperation extended to us and our friends, who helped me directly or indirectly in the successful completion of this seminar.

**NIKITH.H.A (1NH12CS076)**

**INDEX**

1. INTRODUCTION 3-4

2. ALGORITHM 5-9

2A.Active VM algorithm

2B.Problem with current Active VM load balancer

2C. Proposed VM load balancer algorithm

3. EXPERIMENTAL SETUP 10-12

3A.Proposed Algorithm Implementation for cloud Analyst

3B. cloud AnalystSimulation Configuration

4. EXPERIMENTAL RESULTS 13-15

5. CONCLUSION 16

6. REFERENCE 17

**ABSTRACT**

Cloud computing is a growing computing model that is influencing every other entity in the global business industry. Efficient load balancing techniques plays a major role in cloud computing by allocating requests to computing resources efficiently to prevent under/over-allocation of Virtual Machines (VMs) and improve the response time to clients. It is observed that during peak hours when request frequency is high, active VM load balancer (packaged in cloudAnalyst) over-allocates initial VMs and under-allocates later ones creating load imbalance. In this paper a novel VM load balancing algorithm is proposed that ensures uniform allocation of requests to virtual machines even during peak hours when frequency of requests received in data center is very high to ensure faster response times to users. The simulations results suggest that our algorithm allocates requests to VM uniformly even during peak traffic situations.

**1. INTRODUCTION**

Cloud computing is classified as a new paradigm for the dynamic provisioning of computing resources delivered by the state-of-art data centers using virtualization technology. The practice of using a network of remote servers hosted on the Internet to store, manage, and process data, rather than a local server or a personal computer. **Cloud computing**, also **on-demand computing**, is a kind of Internet-based computing that provides shared processing resources and data to computers and other devices on demand. The cloud computing services can be classified into three categories Infrastructure-as-service(IaaS), platform-as-service(SaaS) and software-as- service(SaaS).

Virtualization is an important and core technology for cloud computing. Hardware virtualization or platform virtualization refers to the creation of a [virtual machine](https://en.wikipedia.org/wiki/Virtual_machine) that acts like a real computer with an operating system.It allows the abstraction of fundamental elements of computing such as hardware, storage and networking. Virtualization technology has helped the cloud data centers to effectively increase resource utilization, reduce electricity costs and ease management complexities. But there are many challenges in providing services with reliability and performance .

Load balancing is one of the critical aspect in cloud computing environment that can significantly improve resource utilization, performance and save energy by properly assigning/re- assigning computing resources to the incoming requests from users. Therefore how to schedule virtual machines (VMs) effectively by considering various parameters becomes very important point.

Load balancing is an important issue in parallel and distributed systems. So far, extensive research work has been done to propose various load balancing approaches. Few of the important load balancing algorithms that are proposed by researchers are:- VM resources based on genetic algorithm, Artificial bee colony algorithm weighted least connection(WLC), Load balancing Min-Min algorithm, Equally Spread Current Execution (ESCE) load algorithm.

**2.**  **ALGORITHM**

This section has been divided into three sub parts. The first sub section explains the active VM algorithm,then the problem observed with the current algorithm is explained in the second sub-section and at the end of this section introduces proposed algorithm to overcome the problem.

***2A. Active VM algorithm***

The active VM algorithm allocates the requests to the first least loaded VM at a particular instance of time to maintain uniform allocation for all the VMs. The active VM load balancer is utilized by the data center controller for VM allocation to incoming requests.

**Algorithm: Active VM load balancer**

**Initialization:**

1. **for**( all VM ids)**do**

Set allocation table entries to zero

**end**

Also get the VM status table for all VMs from data center controller.

2.Data center controller requests for VM id to the active VM

load balancer.

**Allocation:**

3.**if** (all VMs are not allocated) **then**

**for** (for all VM ids) **do**

Check for index of allocation statistics table with

zero allocation. If found return VM id.

**end**

**end**

**4.else if** (all VMs are busy) **then**

**for** (for all VM) **do**

Find the index of the allocation statistics table with

least number and return VM id.

**End**

**end**

5.Active VM load balancer recommends a VM id to the data center controller unit.

6. Data center controller assigns the VM id to the request.

7. A notification is sent from data center controller to load balancer about recent allocation for the corresponding VM id.

8. Active VM load balancer increments the count for allocation in the allocation statistics table.

**De-allocation:**

9. When the request completes the processing on VM, data center is notified about completion.

10. Data center then sends the notification to the active VM load balancer to signal VM de-allocation.

11. Active VM load balancer’s allocation statistics table is updated by decrementing the count of requests against the allocated VM id.

***2B. Problem with current Active VM loadbalancer***

The problem with active VM load balancer is observed during peak hours request frequency is very high. During the peak hours, the data center controller queries the active VM load balancer for frequently obtaining VM id for the allocation to the request received . The active VM load balancer returns the suitable VM id to the data center controller at step 5 but it has to wait till load balancer get the notification from the data center controller about the allocation to update its allocation statistic table at the step 8. If any request is received by the load balancer between step 5 and step 8, the load balancer is ignorant about the VM id allocation and allocation statistics table is not updated to reflect the VM ids returned at step 5 by previous requests.

During peak hours, the data center controller experiences high requests frequency and makes frequent calls to the load balancer to request suitable VM for allocation, as the result there are some handling of requests done by load balancer between step 5 and step 8 with allocation statistics table not properly updated. This result in over allocation of initial VM’s as more requests gets assigned to the VM’s listed at the top of the table.

***2C. Proposed VM load balancer algorithm***

A variant of active VM algorithm is proposed to solve the issue during peak hours by using a Reservation table.

The proposed VM load balancer maintains an internal reservation table to maintain the information of VM reservations suggested by the load balancer to data center controller but not updated in allocation table until the notification arrives of allocation. The proposed load balancer takes into consideration both reservations and allocation statistics table entry for particular VM id by the load balancer for VM selection for next request.

**Algorithm: Proposed VM load balancer**

**Initialization:**

1. **for** (all VM ids) **do**

Set allocation table entries to zero **and reservation table**

**entries to zero**

**end**

Also get the VM status table for all VMs from data center

controller.

2. Data center controller requests for VM id to the active VM

load balancer.

**Allocation:**

3. **if** (all VMs are not allocated) **then**

**for** (for all VM ids) **do**

Check for index of allocation statistics table with zero allocation. If found

return VM id.

**end**

**end**

4. **else if** (all VMs are busy) **then**

**for** (for all VM) **do**

**Find the index of the allocation statistics table and reservation**

**table with min\_count(allocation count + reservation count).**

**end**

**end**

5. Active VM load balancer recommends a VM id to the data center controller unit **and updates the reservation table to reflect the allocation against the VM id by incrementing the reservation count**.

6. Data center controller assigns the VM id to the request.

7. A notification is sent from data center controller to load

balancer about recent allocation for the corresponding

VM id.

8. Active VM load balancer increments the count for

allocation in the allocation statistics table **and decrements the reservation table count for corresponding VM id.**

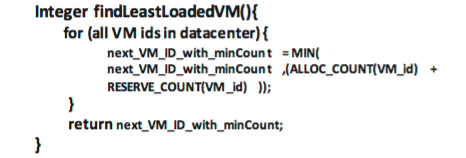
**De-allocation:**

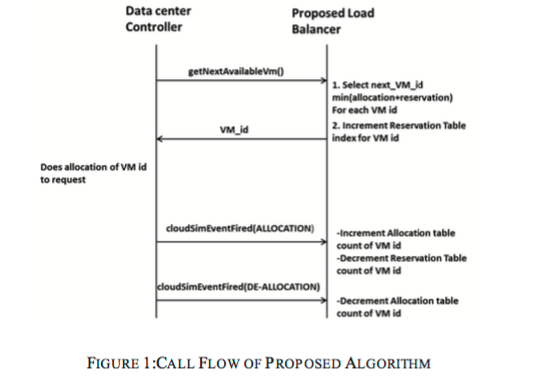
9. When the request completes the processing on VM, data center is notified about completion.

10. Data center then sends the notification to the active VM load balancer to signal VM de-allocation.

11. Active VM load balancer’s allocation statistics table is updated by decrementing the count of requests against the allocated VM id.

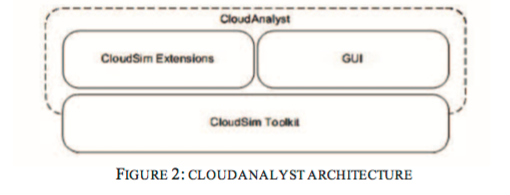
This algorithm takes into consideration both allocation and reservation for VM id to assign the incoming request as shown in pseudo code below

****

****

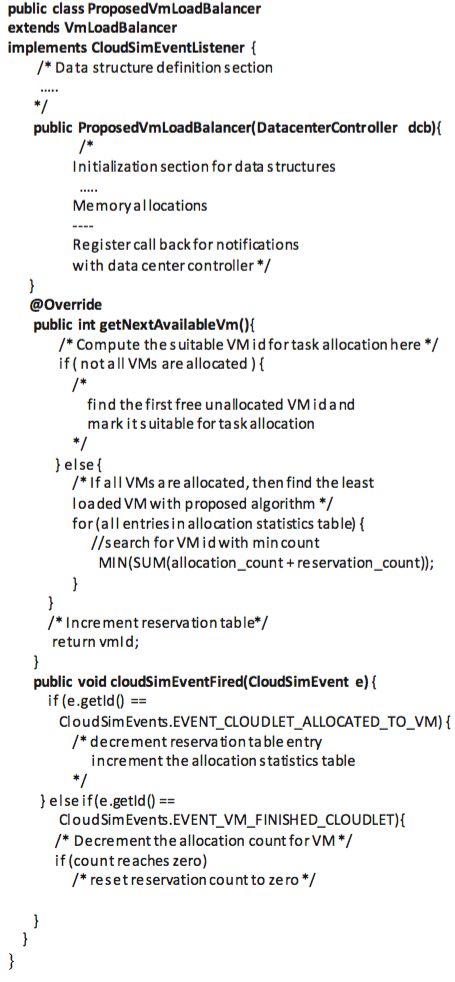
**3. EXPERIMENTAL SETUP**

CloudAnalyst has been used to carry out evaluations of the proposed VM load balancer and compare the result with current active VM algorithm. 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.

****

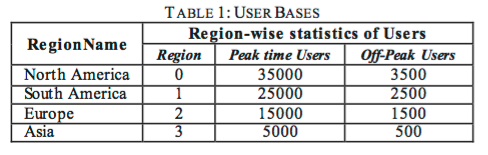
***3A. Proposed Algorithm Implementation for cloud Analyst***

The proposed algorithm is implemented using Java language and integrated into cloudAnalyst source base. The important segments of source code is given in following section for reader’s reference.

****

***3B. Cloud Analyst Simulation Configuration***

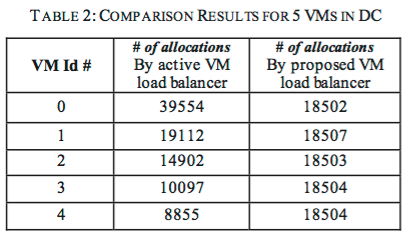
For experimentation, they have considered 4 different regions as continents with different peak and non peak users.in table 1

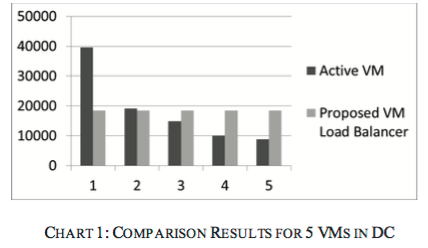


***4*. EXPERIMENTAL RESULTS**

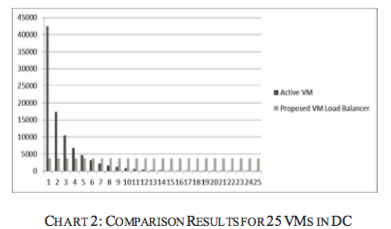
Results are analyzed with major focus on the uniform utilization of the virtual machines by preventing overloading and underloading of VMs and comparing the proposed VM load balancer algorithm with the existing active VM load balancer.

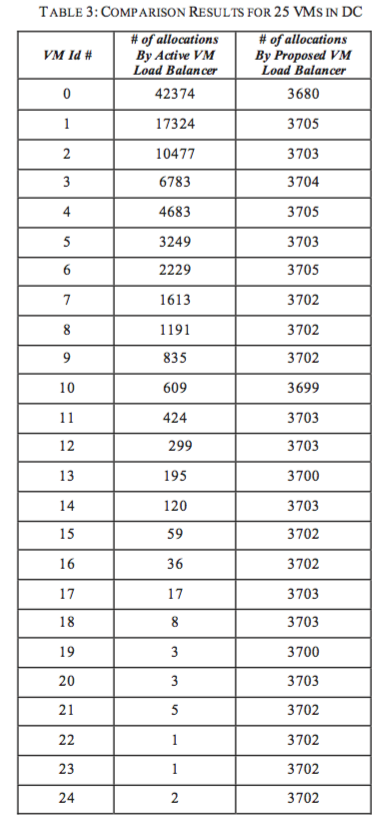
The request allocation numbers of each VM for both current active VM algorithm and proposed VM Load balancer are tabulated in both table 2 and chart 1 for 5 VMs

****

****

The results are tabulated into table 3 and chart 2 for 25VMs and it is noticed that the initial VMs are over loaded and the later ones are under loaded by using active VM load balancer ,where as the result using the proposed algorithm show almost equal distribution across all the VMs



****

**5. CONCLUSION**

In this paper, an efficient VM load balancing algorithm is proposed that distributes the load evenly across all VMs in the data center even during peak hours. It is observed from the experimental results that current active VM load balancer heavily loads the first VM with the arriving requests which is at the top of the table where-as the proposed VM Load balancer equally distributes the incoming requests to all VMs ,using a reservation table between the phase of selection and allocation of VMs. Future enhancement of the project is to investigate the efficiency of applying the technique across all data centers to allocate the load uniformly to all VMs situated at different geographical locations.

**6. REFERENCES**

1. [1]  Ibrahim Takouna, Wesam Dawoud, and Christoph Meinel, “Analysis and Simulation of HPC Applications in Virtualized Dat a Cent ers”, IEEE Int ernat ional Conference on Green Computing and Communications, Conference on Internet of Things, and Conference, 2012.
2. [2]  Jinhua Hu, Jinhua Hu, Guofei Sun, Tianhai Zhao,”A Scheduling Strategy on Load Balancing of Virtual Machine Resources in Cloud Computing Environment”, 3rd International Symposium on Parallel Architectures, Algorithms and Programming.2010.
3. [3]  B.Wickremasinghe, R.N.Calheiros,R.Buyya, “Cloudanalyst: A cloudsim based visual modeller for analy sing cloud computing” in: Proceedings of the 24th International Conference on Advanced Information Networking and Applications (AINA 2010), Perth, Australia,, 2010.
4. [4]  Jing Yao, Ju-hou He,“ Load Balancing St rat egy of Cloud Comp ut ing based on Art ificial Bee Algorit hm” in 8t h Int ernat ional Conference on Comp ut ing T echnology and Information Management (ICCM), 2012.
5. [5]  Ren, X., R. Lin and H. Zou, "A dynamic load balancing strategy for cloud computing platform based on exponential smoothing forecast " in p roc. Int ernat ional Conference on. Cloud Computing and Intelligent Systems (CCIS), IEEE, pp: 220-224, September 2011.
6. [6]  Lee, R. and B. Jeng,"Load-balancing t act ics in cloud," in proc.International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), IEEE, pp:447- 454, October 2011.
7. [7]  Wang, S-C., K-Q. Yan, W-P. Liao and S-S. Wang,"Towards a load balancing in a three-level cloud computing network," in proc. 3rd International Conference on. Computer Science and Information Technology (ICCSIT), IEEE, Vol. 1, pp:108- 113, July 2010.
8. [8]  Sang, A., X. Wang, M . M adihian and RD. Gitlin, "Coordinated load balancing, handoff/cell-site selection, and scheduling in multi-cell packet data systems," in Wireless Networks, Vol. 14, No. 1, pp: 103-120, January 2008.
9. [9]  Ajit M ,Vidy a G,“ VM level load balancing in cloud environment”, Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT), 2013.

**SYNOPSIS**

The proposed algorithm explained in the first section ensures even allocation of requests to all VMs to avoid any over- allocation or under-allocation to any particular VM even during the peak hour traffic situations to improve the response time for the user requests arriving at the cloud data enters. The section is divided into three subparts ,the first part explains active VM algorithm

The active VM algorithm allocates the requests to the first least loaded VM to maintain uniform allocation for all the VMs. The active VM load balancer is used by the data center controller for VM allocation for the incoming requests.

Problems facing this *ActiveVM load balancer* is when the request frequency in data center is very high(peak hour) he data center controller queries the active VM load balancer frequently to get the suitable VM id for the allocation to the received request. The active VM load balancer returns the suitable VM id to the data center controller but it has to wait till load balancer get the notification from the data center controller about the allocation to update its allocation statistic table This will result in over allocation of initial VM, which is at the top of the table.hence the initial VM has more task at hand.

So to prevent this issiue from happening during peak hour by maintaining a Reservation table.Proposed VM load balancer algorithm takes into consideration both reservations table entry and allocation statistics table entry for particular VM id by the load balancer for VM selection for next request. The following proposed algorithm makes sure the algorithm does uniform allocations even during peak hours. CloudAnalyst is used for implementing the current VM load balancer and compare the result.and there for the results shown(table and the chart), the propose algorithm effeiciently distributes load compared to active VM algorithm.