International Journal on Future Revolution in Computer Science & Communication Engineering ISSN: 2454-4248
Volume: 4 Issue: 4 520 – 524

Dynamic Virtual Machine Migration using Network Aware Topology

Dhara Kundariya
Research Scholar
Marwadi Education Foundation
Rajkot, India
pateldhara421@gmail.com

Kashyap Raiyani
Assistant Professor
Marwadi Education Foundation
Rajkot, India
kashyap.raiyani@marwadieducation.edu.in

Tejas Chauhan
Assistant Professor
Marwadi Education Foundation
Rajkot, India
tejas.chauhan@marwadieducation.edu.in

Abstract—Clients of the applications communicate with the services hosted in the VMs. Many applications have clients all over the world. An application is expected to provide faster access and transmission of data to its clients if it is geographically close to its clients, as some of the research work suggests that geographical distance has impact on quality of service (QoS) [1,2,3]. In order to provide a faster access and data transfer, applications which have clients all over the world should be hosted in a data center, which is on average close to its clients geographically.

Keywords—*Virtualmachine,Physicalmachine,Virtualization,Resource management ,Distributed database,Network aware topology*

*****

I. INTRODUCTION

Distributed computing [1, 2] rising as another worldview for next generation figuring in the field of software engineering and data innovation due to their appealing administrations such as simple to utilize, on the web, on request and pay as utilize plot. Cloud is a plan of action, which the on request administrations to the client. Client can get to these administrations whenever at anyplace in the world. Cloud bolster three sorts of administrations i.e. Programming as a Administrations (SaaS), Platform as a Services (PaaS) and Framework as a Services (IaaS) [3, 4]. It can be conveyed in three distinctive way i.e. Private cloud, Public cloud and Hybrid cloud [3, 4]. Private cloud is more secure than general society cloud. The literature survey of papers, Section IV provides the different techniques of machine learning, Section V described tools used to identify the phishing sites. We conclude our work in last section VI that contains some statistics of phishing scenario.

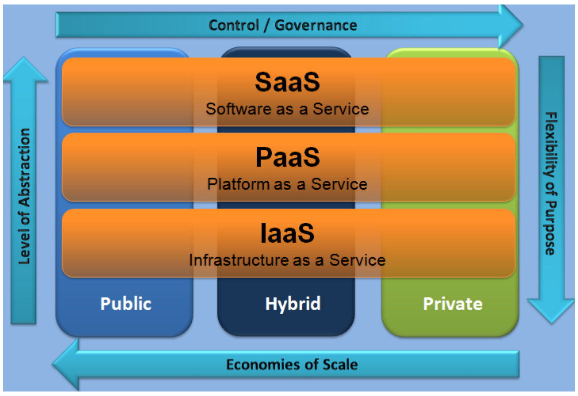


Fig 1 Cloud service model

Virtualization [5, 6] is the center innovation in the cloud, which permits the sharing of the physical assets. With the assistance of virtualization single physical gadget can be share by the various clients. At the point when any client requests for the assets hypervisor or virtual machine screen (VMM) make a VM what's more, tie the asked for assets with the VM. Virtualization can be arranged in two kinds i.e. Full virtualization and paravirtualization. Full virtualization is a strategy in which a finish establishment of one machine is keep running on another machine. In full virtualization, the whole framework is copied (Profiles, drive, et cetera), yet in paravirtualization, its administration module works with a working framework that has been changed in accordance with work in a virtual machine. Para virtualization normally runs superior to anything the full virtualization show, basically since in a completely virtualized arrangement, all components must be imitated. Number of VM can be made in each host and each VM carry on like a physical machine.

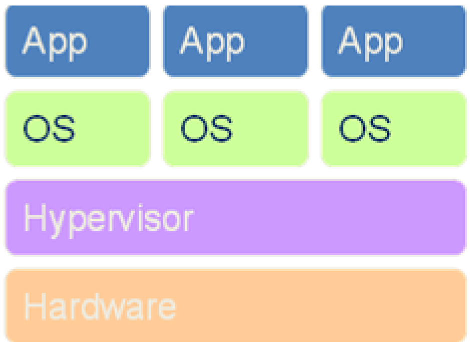


Fig- 2 Virtualization

Assets in the cloud are heterogeneous and topographically circulated. Besides asset request by the client can change powerfully at run time. So the asset administration also, the asset planning for such a huge scale circulated condition is an extremely difficult errand. Assets administration is a center capacity of any powerful frameworks, it requires a few complex approaches and choices for the administration of multi measurement goal, for example, CPU, memory and system data transmission. For the best possible usage of the assets an productive load adjusting methodology are required. Load adjusting approach can be static or dynamic. In the static approach settled limit are utilized that can not changed with time to characterize a level of assets that can be utilized. While in the dynamic stack adjusting approach edge can be changed with time. Static load adjusting approach isn't reasonable for the cloud, where client demand can change with time. By and large lower and upper limit are utilized to characterize the under loaded and over-burden have individually. So in this paper we proposed twofold limit based dynamic load adjusting approach. In the event that stack on the host is underneath the upper edge all VM running on that host are move to the next host, which is known as a server union. On the off chance that heap on the host is more prominent than the upper edge then our host is over-burden along these lines some VM must be relocated. VM relocation [7, 8] procedures are utilized as a part of the heap adjusting. VM relocation is where VM are moves from one host to the another host.

II. Literature Survey

Paul et al. [9] proposed a calculation which is center around how to use asset effectively in distributed computing and pick up most extreme benefits. They regarded undertaking planning as a general task issue to locate the negligible cost. For this reason, they proposed a credit based planning calculation, which assess the whole gathering of errands in the undertaking line and discover the insignificant finish time of all undertakings. The proposed booking strategy considers the booking issue as a task issue in arithmetic where the cost framework gives the cost of an errand to be appointed to an asset. Principle point of this calculation is to dole out the assets for which the relating finishing time of all occupations is the base. Issue with this calculation is that it just considers the likelihood of an asset to be free not long after subsequent to executing an assignment with the goal that it will be accessible for the following pausing, yet preparing time of a vocation are not considered.

Subramanian S. et al. [10] proposed apriority based calculation for the VM booking. In this approach VM are booked as indicated by the need. They allocated some need for each work, which fluctuates powerfully in light of their ability and load factor. This dynamic need idea prompts better use of the assets. Needs are doled out in light of the measure of assets required in each measurement and volume. Higher need are allocate to the VM which having a higher asset prerequisite. This calculation adjust the asset however it will increment the sitting tight time for the VM which having a lower need.

Mayank Mishra et al. [11], Proposed a technique for setting the VM which depends on Vector hypothesis. They are utilizing asset vectors TCV, RUV, RCV and RRV in the 3-D space which we will use for settling on various VM arrangement choices. One of our prime objectives while setting VM is to make the asset use of PMs as adjusted (along every asset measurement) as could be allowed, i.e., the RUV of a PM ought to be as firmly adjusted to the TCV as could be expected under the circumstances. This would require that we have a way of discovering correlative VM for a PM. They legitimate adjust the assets however not center to the server solidification. So vitality devour by the server farm is high.

T. Wood et al. [12], proposed an approach for the problem area relief know as sandpiper. Sandpiper utilize dark and dim box way to deal with screen the host. They utilize the Xen hypervisor. The checking motor is in charge of following the processor, system and memory use of each virtual server. It likewise tracks the aggregate asset use on each physical server by totaling the utilizations of inhabitant VMs. Issue with this approach is that they just consider the cpu load to compute the heap on have.

A. Beloglazov et al. [13], proposed a vitality productive load adjusting approach. They contend that normal power devoured
by a sit without moving server is 70% of energy devoured by completely used server. So control devoured by the server farm can be controlled by the correct load adjusting approach. They utilized settled lower and upper edge with the distinction of 40 amongst lower and upper edge. So if bring down limit is 30 than upper edge is 70. This approach decreased the number of relocation yet fundamental issue with this approach is that they utilized the settled estimation of lower and upper edge.

III. Proposed Framework

A cloud domain comprises of server farm, VM and host. Every datum focus can have various hosts and each host can run number of VM. At the point when client requests for the assets VMM make a VM and dole out to the client. VMM is a fundamental piece of the virtualization, which handle all VM related undertaking. So VM creation, erasure and planning all are finished by the VMM. It is likewise in charge of the checking of the assets, for example, CPU, RAM utilized by the VM and PM. Asset usage can be expanded by the virtualization however for the correct use of the asset a proficient load adjusting approach is required that take the choice as indicated by the circumstance. Here we proposed twofold edge based dynamic load adjusting approach, where edges are figured in view of the host usage. On the off chance that heap on the host is underneath the upper limit all VM running on that host are move to the next host, which is known as a server union. On the off chance that heap on the host is more prominent than the upper edge then our host is over-burden thusly some VM must be relocated. At the point when the heap on the host is beneath/over the edge then VM relocation system are

ISSN: 2454-4248
520 – 524

utilized. It is where VM is move from one host to another host. Amid the movement VM is suspended for a couple times, that will diminish the execution of the framework. So an viable load adjusting methodology ought to limit the number of movement. Principle goal of our approach is to diminish the number of relocation.

Four stages are associated with the VM movement.

i. Ascertain stack on the PM and VM.
ii. Compute the upper and lower edge to discover the over-burden and undreloaded condition.
iii. Select the best VM for the relocation
iv. Select the best host to put the chose VM

Load Calculation for the Physical and Virtual Machine

We consider three parameter i.e. CPU, memory andbandwidth for the heap estimation. So each VM have itown CPU, memory and data transmission. Load on the VM can be figured as

$$VM_{cpu} = \frac{\text{totalRequestedMips}}{\text{total MIPS of the PM}}$$

$$VM_{bw} = \frac{\text{Bandwidth use by the VM}}{\text{Total bandwidth of the host}}$$

$$VM_{ram} = \frac{\text{ram use by the VM}}{\text{Total ram of the host}}$$

Load on the VM is depends on the CPU utilization. So load of the VM is directly proportional to CPU utilization and define as a

$$VM_{load} == \frac{\text{totalRequestedMips}}{\text{total MIPS of the PM}}$$

Add up to stack on the host is the aggregate heap of the VM running into that host. On the off chance that there are n VM on path have then normal load on the path host can be figured by given equation

$$PM_{load} = \frac{\sum_{i=1}^{n} VL_i}{n}$$

Lower and Upper Threshold Calculation :

Two edges i.e. lower and upper are utilized to characterize the over-burden and under loaded have. These edges can be Static and dynamic. In the static edge lower and upper limits are settled and they not changed with time, while in the dynamic limit, lower and upper edges are changed with time. Dynamic limit is more appropriate for the cloud, where assets required by the VM are changed powerfully. It is break down that VM movement probability increment with the limit. That implies as the upper limit increment it will additionally increment the potentially of the relocation. A large portion of the work done just thought about the CPU for computing the heap, yet Slam is the most basic component in the framework as contrast with the CPU. So to calculate the upper edge we consider CPU, RAM and transfer speed with parallel weight

$$T_1 = \frac{\sum_{i=1}^{n} VM_i^{CPU}}{\text{Total MIPS capacity of host}}$$

$$T_2 = \frac{\sum_{i=1}^{n} VM_i^{RAM}}{\text{Total RAM of the host}}$$

$$T_3 = \frac{\sum_{i=1}^{n} VM_i^{BW}}{\text{Total bandwidth of the host}}$$

$$temp \leftarrow (T_1 + T_2 + T_3)/3$$

$$T_{upper} = 1 - x * temp$$

$$T_{upper} = 1 - 5 * temp/100$$

$$T_{lower} = 0.03$$

Where n is the quantity of VM in the host and x is the rates of the temp. In light of the test x=.05 is the appropriate esteem, which keep up the tradeoff between the number of movement and assets wastages. Edge for the following interim is figured in view of the past history i.e. limit for the t2 interim relies upon the host usage in the t1 interim.

Virtual Machine Selection:

Each host can have number of VM. So which VM is chosen for the relocation influence the aggregate movement time and down time. Down time is the ideal opportunity for which VM not accessible to the client furthermore, add up to movement

1: Input: hostList, vmList Output:migrationList

2: Mastermind each host into diminishing request of their use

3: for every h in hostList do

4: hostUtil ←host.util()

5: bestVmUtill ← Utilization of first VM

6: while hostUtil > host.upThresh do

ISSN: 2454-4248
520 – 524

7: for each vm in vmList do

8: diff ← hostUtil ─ host.upThresh

9: on the off chance that vm.util() > diff at that point

10: temp ← vm.util()

11: on the off chance that temp < bestVmUtill at that point

12: bestVmUtill ← temp

13: bestVm ←vm

14: else

15: on the off chance that bestVmUtill = First VM at that point

16: bestVm ← vm

17: break

18: hostUtil ← hostUtil ─ bestVm.util()

19: migrationList.add(bestVm)

20: vmList.remove(vm)

Select the Host to Place the Selected VM :

Target PM Selection is the most basic advance in the VM relocation, since it influences the general execution of the framework. Wrong determination of the PM may build the quantity of VM relocation and also asset wastage. In our approach we select the VM which is control productive.

1: Inputs: hostList, vmList Output: designation of VMs

2: Sort all PM agreeing their usage

3: foreach vm in vmList do

4: foreach have in hostList do

5: on the off chance that Host_Load<=H_UTD && Host_Load>=H_LTD

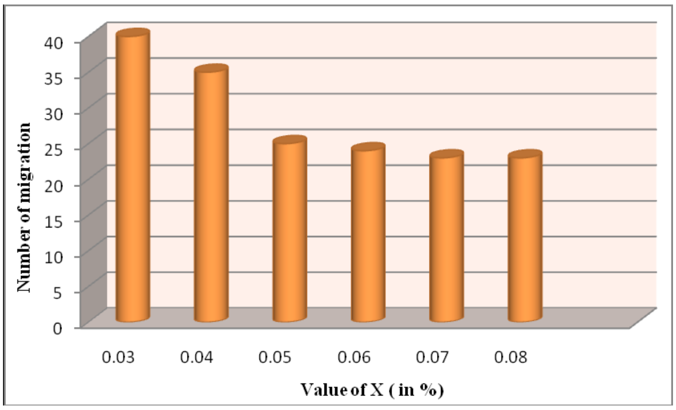
6: Relegate VM to the host where less augmentation in the power

7: else

8: Initiate new host and place out VM to that host.

IV. Experiment Result

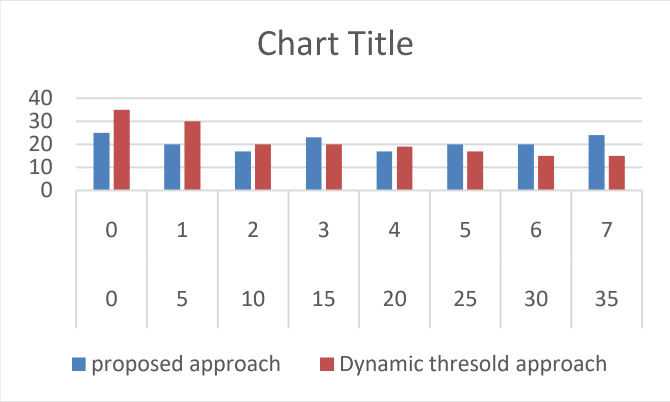
For calculating different value of X



To execute our approach we are utilizing CloudSim test system
[14]. This test system depends on the Java and contained the classes, for all the capacity which is required to execute the cloud based methodologies. To check the proficiency of our approach we contrast our calculation and the leaving vitality mindful asset distribution approach [13]. We have reenacted a server farm 10 PM. Every PM is displayed to have one CPU center with the execution proportional to 1000, 8 GB of RAM and 1 TB of capacity. Greatest power devour ed by the host is 250 W. So as indicated by the power display [13], a host expends 175 W with 0% CPU usage, up to 250 W with 100% CPU usage. Each VM requires one CPU center with 250 MIPS, 128 MB of RAM and 1 GB of capacity. Each VM runs a few application with 150,000 MI, which required 10 min on the 250 MIPS with 100% use. To produce the variable load auniform arbitrary capacity is utilized, which produce theirregular esteem between 1-250. Since asked for MIPS by theVM changed without fail, so for every execution it gives the unique number of movement for a similar number of physical and virtual machine. In this way each analysis has been run 10 times.

At first we take 10 hosts and 20 VM. To ascertain the upper limit we execute our approach 20 times for the extraordinary estimation of X, where is the rate (X=.03 to .08) and ascertain the normal number of movement. We found that x=.05 is the best esteem.

ISSN: 2454-4248
520 – 524



Total energy consume by the data center is depends on thenumber of migration. So we plot the graph between the time and number of migration.

V. Conclution

Load balancing is a very important task in every system, because the system performance is totally depends on the load management. But load balancing in cloud is very challenging task, due to the resource required by the VM is changed dynamically. Furthermore resources in the cloud are distributed dynamically. In this paper we proposed a double threshold based load balancing approach. In this approach VM migration approach is used to balance the system. Lower threshold are use to implement the concept of server consolidation and upper threshold are use for the load balancing. Experiment result show that our approach reduced the number of migration as well as energy consumption.

References

[1] M. Armbrust et al., “Above the Clouds: A Berkeley View of Cloud Computing”. In: EECS Department, University of California, Berkeley (2009)Rajyashree, IJECS Volume 4 Issue 1 January, 2015 Page No.9966-9970 Page 9970

[2] RK Gupta et al., “A Complete Theoretical Review on Virtual Machine Migration in Cloud Environment”, IJ-Closer,vol. 3, pp. 172-178, 2014.

[3] V. Sarathy et al., “Next generation cloud computing architecture”, 19th IEEE int. workshop on enabling technology:infrastructure for collaborative enterprise, pp.48 -53,2012.

[4] R. Buyya et al. Market-oriented cloud computing: Vision,hype, and reality for delivering it services as computing utilities, Proc. of the 10th IEEE Intl. Conf. on High Performance pp. 54-58, 2011.

[5] Borja Sotomayor et al., “Enabling cost-effective resource leases with virtual machines,” Research Gate article may 2014..

[6] L. Cherkasova et al. “When virtual is harder than real: Resource allocation challenges in virtual machine based itenvironments” in proc. 10th conference on hot topic in operating system, Vol. 10, pp.20-20, 2005.

[7] H. Jin et al., Live virtual machine migration withadaptive memory compression” proceeding of the IEEEinternational conference on cluster computing, pp. 1-10, 2012.

[8] H. Jin et al., “Live migration of virtual machine based onfull system trace and replay”, proceeding of the 18th ACM2009.

[9] Paul, M. et al. “Task-scheduling in cloud computing usingcredit based assignment problem”,.International journal ofComput. Sci. Eng., pp. 26-30, 2011.

[10] Subramanian S et al., “An Adaptive Algorithm ForDynamic Priority Based Virtual Machine Scheduling In 2 Cloud”IJCSI International Journal of Computer Science Issues, Vol. 9,Issue 6, No 2, November 2012.

[11] Mayank Mishra et al., “On Theory of VM Placement:Anomalies in Existing Methodologies and Their MitigationUsing a Novel Vector Based Approach”, IEEE/ACM 4thinternational conference on cloud computing, pp 275-282, July2011.

[12] T. Wood et al., “Black-Box and Gray-Box strategies forvirtual machine migration”, NSDI'07 Proceedings of the 4thUSENIX conference on Networked systems design & implementation, pp. 7-17, 2007.

[13] A. Beloglazov et al. “ Energy-aware resource allocationheuristics for efficient management of data centers for Cloudcomputing”, Elsevier journal of Future Generation ComputerSystems, pp. 755–768, 2012.

[14] R.N. Calheiros et al., “CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms, Software: Practice and Experience”, pp. 23–50, 2011.