CLOUD ENVIRONMENT WITH THRESHOLD BAND BASED FOR AUTOMATIC LOAD BALANCING

Siva Naga Manikanta Immidi[1], K Durga Prasad[2], Prof.S.V.Achutha Rao[3]

[1]*M.Tech (CSE) Scholar Vikas College of Engg and Tech., Nunna, Vijayawada, A.P, (India)*

[2]*Asst. Professor, Department of Computer Science Engg,*

Vikas College of Engg and Tech., Nunna, Vijayawada. AP, (India)

[3]*Professor & Head, Department of CSE,*

Vikas College of Engg and Tech., Nunna, Vijayawada, (India)

ABSTRACT

Virtualization is used in Cloud computing shares the essential hardware and it offers computing resources through virtual machines (VM). VMs provide separation among individual OSs. Resource usage could be upgraded by balancing the loads on the cluster and which can be reached through dynamic migrations of virtual machines anywhere required. Physical load balancing becomes very monotonous, meanwhile an administrator has to endlessly monitor the system to control which virtual machines to be transferred across the cluster. Maximum source use can be reached using involuntary load balancing through the cluster by doing active migrations of virtual machines and it is very important in such type of migrations of virtual machines to decide when to move and which virtual machine to what physical machine. Naive immigrations may result into ruin in overall performance of the cluster. The recommended solution encompasses a model which dynamically controls overloaded hosts and picks the finest virtual machine and endpoint host for migration to achieve improved load balanced environment

Keywords- Threshold Band, Load Balancing, Cloud Computing, Virtual Machines

I. INTRODUCTION

Cloud computing is a approach for enabling suitable, on demand network access to a shared pool of configurable calculating resources like , storage, networks, applications, servers, and services that can be quickly provisioned and released with insignificant organization effort or service provider interaction. Many types of services in Cloud Computing like Software as a Service, Public as a Service, and Infrastructure as a Service etc. Cloud Computing depends on Virtualization for service implementation and distribute resources to the end users over the web to achieve scalability. Virtualization presents a software abstraction layer among the hardware and the operating system and applications running on top of the abstraction layer. It gives full control of resource allocation to manager, resultant in finest use of assets. Load balancing is the procedure of improving the performance of the system by shifting of workload among the all processors. Rapid migration of virtual machines is a load balancing method which allows us to attain best use of offered resources dynamically in a cloud computing environment. In the method presented by Hyung Won Choi involuntary load balancing is accomplished by the use of antiquity matrix which keeps records of previous migrations situations to predict the upcoming decisions. The approach claims to perform better than the approach with threshold based migrations. In our basic proposed approach, we recommend a model which will take well-organized choices for migration of virtual machines from the burdened hosts. The choices depend mostly upon 3 things like CPU consumption of all hosts called as the threshold value, main eccentricity of CPU consumptions of hosts around the threshold rate and the CPU utilization of the most overloaded host.

1.1 Load Balancing Concepts

1.1.1 Virtualization

In cloud computing, there is a requirement to create a virtual environment by separating resources like memory, applications, CPU, network and storage. Virtualization attains this with software as a service based implementation and also the end user can cooperate with the computer-generated framework like a real single consistent resource and it hides the physical arrangement of the original resource and in its place it shows another abstract computing platform. It is elastic in terms of providing flexible provisioning of the services in a timely and on-demand manner, which allows the cloud users to flexibly scale up and scale down the resources as per the need. There are mainly two leading ways of implementing virtualization

1.1.2 Full Virtualization

Virtual Machine Monitor (VMM) is Virtual Machine Manager which accomplishes multiple operating systems by distributing essential resources and preserving isolation between every virtual machine. VMM offers an identical abstraction of the primary hardware to the virtual machines in full virtualization. But, not all designs are virtualizable. Full virtualization totally separates primary hardware from the software and in that virtual environment; the guest operating system need not to be changed. It will run also on the virtualized environment as it would run in physical system.

1.1.3 Load Balancing

Well-adjusted systems always result in well performance and energy preservation. Load balancing is becoming a very significant consideration for all cloud service providers and data canters. In order to attain minimum resource consumption, avoid blocks, upgraded fault tolerance, scalability and over-provisioning. To gain enhanced computing resources, burden is distributed across all the host systems in the cluster. Load balancing allocates even workload to avoid conditions where some servers are heavily loaded and some are less loaded or idle. The main goal of load balancing is to decrease resource consumption and to attain less response time by preserving the balanced environment. Load balancing procedure runs for every cluster and controls source and target for relocation to gain better usage of assets.

II. RELATED WORK

In cloud environment many virtual machines are organized on a cluster of hosts and also the workload on every virtual machine could be random and also it may be change dynamically, which will leads to attention of load on some hosts. If the burden increases on a host then the response time of that specific host gets degraded and the cloud service providers need to deal with the Service Level Agreements those are signed with the clients on the response time and also need to exploit the resource consumption in a cloud environment. To meet these requirements, dynamic migrations of virtual machines across different hosts in the cluster are must be required.

To meet these things, mainly 3 tasks should be performed: Identifying overloaded hosts in the cluster, choosing the finest virtual machine for migration, avoiding undesirable migrations. The main aim of the algorithm is separated into 2 parts. (i) To define the hosts which are already overloaded in the cluster, (ii) To decide the finest virtual machine to migrate from these already overloaded hosts and also to specify the destination host for data migration. Algorithm ONE, identifies the overloaded hosts in the cluster and the remaining part, Algorithm TWO, resolves the best virtual machine to migrate from the already overloaded hosts and also the endpoint host for migration.

2.1 Algorithm ONE: Detection of Overload by Using Threshold Band

When the use of a physical machine is beyond a static threshold, the machine is considered as overloaded. This static threshold can become dangerous to total performance of the cluster. Even a single mistake in expecting this threshold value may lead to needless migrations in the cloud. Later, detection of those thresholds need much time, administrator's continuous interference and also the resulting threshold may not be work properly for different kind of situations since the computing loads change over time that's why we propose a new parameter in this algorithm that is called as threshold value of CPU usage of all the hosts in the cluster is denoted by θ. Threshold value is considered occasionally and it is used as a standard to assume a host as overloaded and the hosts which have CPU usage near to this proposed threshold value is named as overloaded. This "immediacy" is enumerated with a model of threshold band. This defined threshold band is a section above the threshold value and the hosts, whose CPU usage lies above this threshold band, is named as unstable. The hosts underneath this threshold band are possible endpoints for migration. Fig 1 specifies the concept of threshold band.

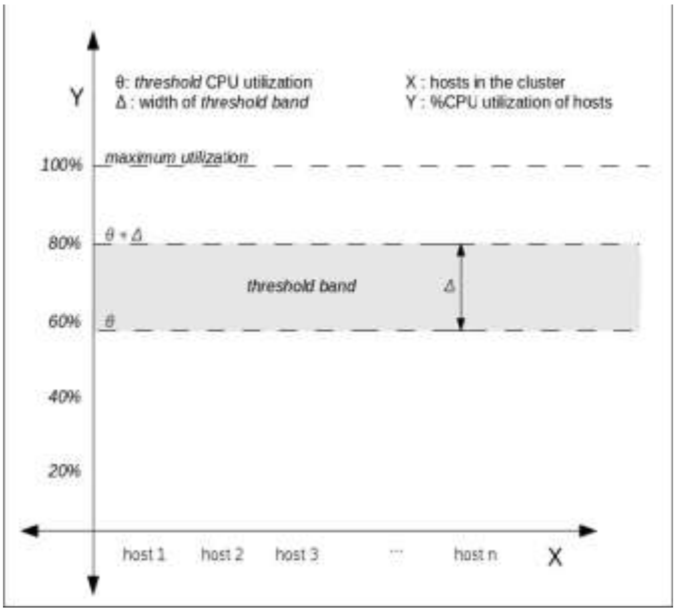

Fig. 1 Threshold Band Concept

2.2 Algorithm ONE

1: Get CPU utilization (αi) of each host i in the cluster

2: Calculate Threshold: $\theta = (\alpha 1+\alpha 2+...+\alpha n) / n$

3: Calculate the Multiplicative factor:

$m = (1 – \theta / 100) + (1 – \alpha max / 100)$

4: Calculate the Mean Deviation:

$$\omega = 1/n \sum_{i=1}^{n} | \alpha i - \theta |$$

5: Calculate the width of the threshold band:

$$\Delta = m\omega$$

6: Initialize *LIST overloaded* with an empty list

7: **for** i = 1 **to** i = n **do**

8: **if** $\alpha i > (\theta + \Delta)$ **then**

Append αi to *LIST overloaded* (List of overloaded hosts)

9: **end if**

10: **end for**

12: **if** *LIST overloaded* is empty **then**

13: Sleep for 1 second

14: Go to Step 1 of Algorithm ONE

15: **else**

16: Go to Step 1 of Algorithm TWO

17: **end if**

2.2 Algorithm TWO

1: Sort the *LIST overloaded* in descending order

2: *SOURCE load* Select MAX from (*LIST overloaded*).

3: Calculate the current standard deviation of the loads (*SDcurr*)

4: **for all** *VMi* in SOURCE **do**

5: *SOURCE load = SOURCE load −VMi*

6: *αmin = αmin +VMi*

7: *Calculate SDi*

8: **end for**

9: Find Minimum Standard Deviation *SD min* from All *SDi s*

10: **if** *SD min* < *SD curr* **then**

11: Perform Migration of the corresponding *VMi*

12: Go to Step 1 of Algorithm ONE

13: **else**

14: Remove the highest overloaded host from *LIST overloaded*

15: Sleep for 1 second

16: Go to Step 2 of Algorithm TWO

17: **end if**

III. RESULTS

To examine our application we have created a scenario which faithfully relates to a representative computing environment. Table ONE show the sharing of the virtual machines on the hosts and also load on virtual machines. Here we have taken three hosts named as host1, host2 and host3, hosting virtual machines vm1 (Virtual Machine) to vm13 (Virtual Machine). Third column in the table ONE specifies the applications run on every virtual machine. The number after the application name represents the number of occurrences of that application running on that virtual machine and after execution of the loads on every virtual machine reports the total time that is required by virtual machine to execute all the application occurrences allocated to it, to the main center host.

Host	Virtual Machine	App used for loading
host1	Vm10	adpcm(1), gsm(3)
	Vm11	rsynth(1), qsort(1)
	Vm12	fft(5)
	Vm13	basicmath(2)
host2	Vm7	bitcount(4)
	Vm8	fft(5)
	Vm9	sha(1)
host3	Vm1	bitcount(1), basicmath(1)
	Vm2	pgp(2), qsort(4)
	Vm3	dijkstra(5)
	Vm4	susan(1), rsynth(4)
	Vm5	qsort(2)
	Vm6	sha(1)

Table ONE: Load Assigned to the Clusters

We have applied the above table loading scenario and examined the performance of our effective migration algorithm and after that we compare performance of our proposed algorithm with other naïve migration algorithm with nearly 50% threshold value of resource usage and to compare both the procedures, in the part of our experimental results we run the migration algorithm with the same load scenario that is stated in the Table ONE and catch the total execution time of the applications reported by every virtual machine.

IV. CONCLUSION

Dynamic migrations of Virtual Machines assure maximum source usage and lowest reply time in a cluster. We have offered a threshold band based approach which is automatically decides, on the base of the CPU usage, which virtual machine to be migrated to which host and we took the average CPU usage of all the hosts in the main cluster and also the CPU usage of the highest overloaded host, that will be into consideration for scheming the threshold value. Based on the results we accomplish that using our proposed efficient migration algorithm executes 32% faster than the previous naive algorithm for that the solution that we proposed in this paper must be compared with the previously existing solutions. Determining the already existing solutions and comparing with the proposed solution with them, is the future work. The future plans include: 1. including a learning framework for the calculation of CPU usage of all Virtual Machines, 2. testing our application by deploying it on actual cloud with actual life loading environments, 3. removing the disadvantage of failure in the main server by reproducing it to a main backup server.

REFERENCES

[1] Jyotiprakash Sahoo, Subasish Mohapatra, and Radha Lath, Virtualization: A Survey On Concepts, Taxonomy And Associated Security Issues, in Second International Conference on Computer and Network Technology, 2010.

[2] Sandeep Sharma, Sarabjeet Singh, Meenaksshi Sharma, Performance analysis of load balancing Algorithms, in world academy of science in Engineering and Technology, 2008

[3] Gerald J. Popek University of California, Los Angeles ,Robert P. Goldberg Honeywell Information Systems and Harvard University Formal Requirements for Virtualizable Third Generation Architectures

[4] Lamia Youseff1 , Rich Wolski1 , Brent Gorda2 , and Chandra Krintz1 , Paravirtualization For HPC Systems.

[5] Gary chen,Al Gillen,KVM for server virtualization:An open source solution comes on AgeOctober : 2011.

[6] Nidhi Jain Kansal, Inderveer Chana, Existing load balancing techniques in cloud computing: A systematic re-view ,February 15, 2012.

AUTHOR PROFILE

Siva Naga Manikanta Immidi, pursuing M.Tech(CSE) Vikas College of Engineering and Technology, Nunna, Vijayawada. Affiliated to JNTU, Kakinada, A.P., India

K Durga Prasad, working as an Asst. Professor at Vikas College of Engineering and Technology, Nunna, Vijayawada, Affiliated to JNTU, Kakinada, A.P., India

Prof S.V.Achutha Rao, is working as a HOD of CSE at Vikas College of Engineering and Technology, Nunna, Vijayawada, Affiliated to JNTU, Kakinada, A.P., India