International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Impact Factor (2012): 3.358

Flag Based VM Allocation Technique for Load Balancing

Hiren Parmar[1], Tushar Champaneria[2]

[1]L. D. College of Engineering, Gujarat Technological University, Gujarat, India

[2]Assistant Professor, L. D. College of Engineering, Gujarat Technological University, Gujarat, India

Abstract: ***The greatest platform for area of computing and research today and which involves term like Virtualization, Distributed Computing, Networking, Software, and Web Services. Cloud Computing includes Scalability, Flexibility, High Availability, Fault Tolerance, Less Overhead for Users, Less Cost for Ownership, On Demand Services, etc. Central to these issues is how to manage Load among multiple virtualized resources and leads to use effective Load Balancing Algorithm. The Load can be defined by CPU (Processor) Load, Memory, and Network Load. So, what is load balancing??? The Answer is – Distributing the load across various nodes for best resource utilization and best response time and also avoids the situation where some nodes are heavily loaded while other nodes idle. It also ensures that work distribution among all the nodes is exact at any time. In this paper our proposed algorithm is based on frequency base distribution of VM load and according to that we divided total load in to the four parts. If the frequency of the present status is same as the current status the load table will not updated else the load table will change VM status.***

Keywords: Load Balancing, Adaptive Load, Controller Node, Load Table

1. Introduction

NIST definition, "cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction."[1]

Figure 1 : Cloud Services

IaaS Definition Infrastructure as a service (IaaS) is a standardized, highly automated offering, where compute resources, complemented by storage and networking capabilities are owned and hosted by a service provider and offered to customer's on-demand. Customers are able to self-provision this infrastructure, using a Web-based graphical user interface that serves as an IT operations management console for the overall environment. API access to the infrastructure may also be offered as an option.[2]

PaaS Definition A platform as a service (PaaS) offering, usually depicted in all-cloud diagrams between the SaaS layer above it and the IaaS layer below, is a broad collection of application infrastructure (middleware) services (including application platform, integration, business process management and database services). However, the hype surrounding the PaaS concept is focused mainly on application PaaS (aPaaS) as the representative of the whole category.[3]

SaaS Definition Gartner defines software as a service (SaaS) as software that is owned, delivered and managed remotely by one or more providers. The provider delivers software based on one set of common code and data definitions that is consumed in a one-to-many model by all contracted customers at anytime on a pay-for-use basis or as a subscription based on use metrics.[4] More and more people pay attention to cloud computing. [5, 6] In cloud computing, there will be lots of machines for the purpose of computation. So, there is a need that each and every machine is equally loaded.

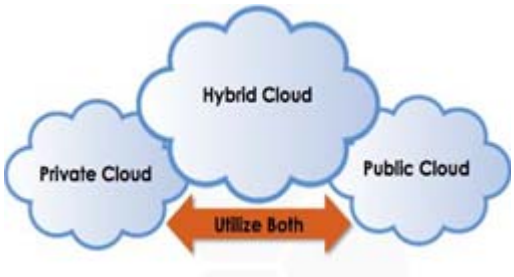

Figure 2: Cloud Models

There are basically three types of cloud: public cloud, private cloud and hybrid cloud.
Public Cloud is the standard models for cloud computing where storage, services, application are on the internet for 'as a services' – pay per use models.
Private Cloud is the single organization infrastructure where both IT department and management to virtualized the requirement of business.
Hybrid Cloud, suppose the company is not limited to the single location and have different branches across the world. Then if all the branches have their own infrastructure with the help of private cloud but they can connect with the help of public cloud. It is the combination of both the cloud models
There are basically two types of load balancing algorithm: Static and Dynamic. The static algorithm is easily come into the execution and will take small amount of time, which also doesn't depend on the present states of the server nodes. Some of the static algorithm are Random [7], Round-Robin

[8], Weighted Round-Robin [9], etc. But, Dynamic algorithm depends on the present states of the server nodes. Some of the Active Monitoring [10], etc. The article is aimed at the private with Dynamic Algorithm cloud which has number of hosts with distributed computing resources. This model will have one datacenter broker which will submit request to the Hosts and ultimately to the VM. As VM receives request it will have some of the load on it. And based on the present condition of VM we will generate different kind of flags based on frequency distribution parameter.

2. Related Work

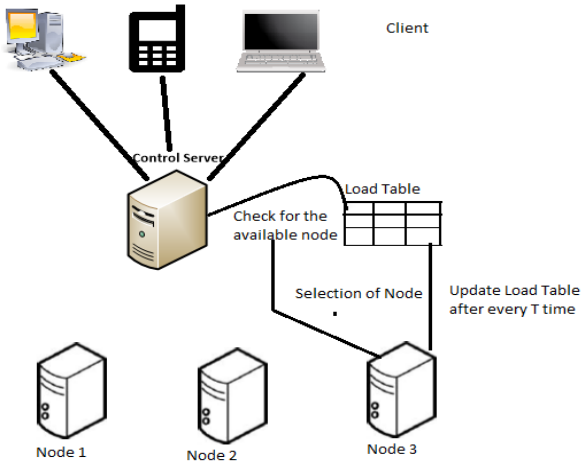

Figure 3: Generic Scenario of Base Paper Algorithm

Gaochao Xu, Junjie Pang, and Xiaodong Fu[11] wrote the paper based on load balancing model based on cloud partition. In this paper, the overall cloud infrastructure divided in to the multiple partitions. When new request arrives it will arrive at the main controller. Main controller will have the statistics about the different partition and based on the statistics it will select the best partition among all. So, every time whenever new request passes to the partition it will generate load on particular partition and that load needs to be updated on the main controller. For every single request main controller and partition will send statistics about its current status and generate network traffic issues.

Gowtham Kanagaraj, Naveen Shanmugasundaram And Sathish Prakash[12] adaptive algorithm for load balancing. In this algorithm, whenever client makes a request from any device it will pass through the central node. Central node has the state information about the different server. The serves are located on server farm. Based on the state information of different server select best efficient server for processing the request. In Adaptive Algorithm, every time there is a requirement for the current state of server. If there are lots of request coming from the different client to the central node and passing that request to the different server and keeping table at central node up to date is rather difficult task. And result in the inconsistency of the node table.

Vikas Patel [13], communication aware approach, three parameter taken Network, CPU, and Memory. This all three parameter is given appropriate weight. Whenever new request arrives it will first calculate the weight, and select the node with maximum weight and consider it as the most powerful machines. But every time when the request is processed under the node it is necessary to calculate the utilization factor of Network, CPU and Memory. Now, after having utilization factor every time new weight table is generated.

Meenakshi Sharma, Pankaj Sharma, Dr. Sandeep Sharma [14], overall algorithm is divided into the three parts, first it will calculate the expected response time of each VM, in second phase will find out the best VM, and in last phase returns the ID of the best efficient VM.

3. Proposed Algorithm

In all above algorithm it has been found that they will only focus on balance the nodes but they don't have accurate load table. Also, they try to have the balanced load table when request passing from the central node to the computing node. But updating the table is very tedious task. For every single change in VM status the load table needs to be updated. Sending and receiving each and every status will generate the network traffic and due to that traffic rather having good load balancing strategy we loss most of our network bandwidth in updating the central node table.

Here, we suggest rather keeping table up-to date for each and time passing the request to central node we will only see the four set of frequency. The overall status is divided in to the four different parts and if the present VM status fits in to the criteria it will send update status to the central node.
Steps for the algorithm is as follows:

Step 1 : Client Makes the request
Step 2 : Central Node Receives that request and will check the load table for passing the request. Initially all the nodes is in the idle condition.
Step 3 : Calculate the Load based on Network, Processor and Memory Parameter and take average of all three.
NU = Network Utilization
PU = Processor Utilization
MU = Memory Utilization

$$Load = \frac{NU + PU + MU}{3}$$

Step 4:
- Idle : when Load = 0 or Load <= 5
- Less Loaded : when Load >5 or Load <=30
- Medium Loaded : when Load > 30 or Load <=70
- Heavily Loaded : when Load > 70

Step 5 : Check for the previous status of the VM in load table if the previous status is same as the present status there is no need to update flag in the table else change it status in to the appropriately.
Step 6 : Go the step 1 until all the request finished
Total number of queue create here will be the four queue based on four status (idle, partially loaded, medium loaded, heavily loaded). Based on status VM id will go in the queue. Now we will select randomly any of the VM from the queue. If the selected VM change its status after passing request we will remove that VM id from the queue and add it to the another queue.

3.1. Scenario 1 : Idle Condition

Load is 0 or less than equal to 5

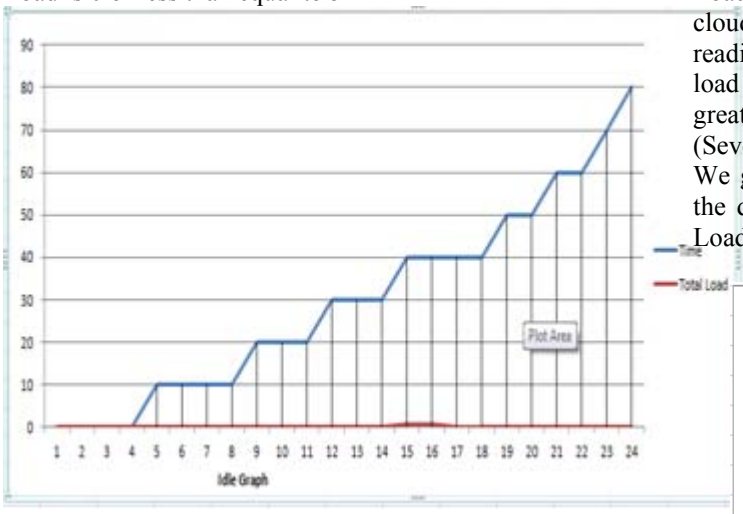

Figure 4: Idle Condition Graph

We send cloudlet to the different VM according to that we took reading about the present status of the VM and if VM total load calculated based on (Network, Processor and Ram) is 0 (Zero) or less than equal to 5 (Five) the status table will mark it as Idle VM. We generate the graph based time versus load parameter on the different situation. Figure 4 shows the graph with idle condition. If idle queue is not empty the request will process from the list of VM under the Idle queue.

3.2. Scenario 2 : Less Loaded Condition

Load is greater than 5 or less than equal to 35. We send cloudlet to the different VM according to that we took reading about the present status of the VM and if VM total load calculated based on (Network, Processor and Ram) is greater than 5 (five) or less than equal to 35 (Thirty-Five) the status table will mark it as Less Loaded VM. We generate the graph based time versus load parameter on the different situation. Figure 5 shows the graph with Less Loaded condition

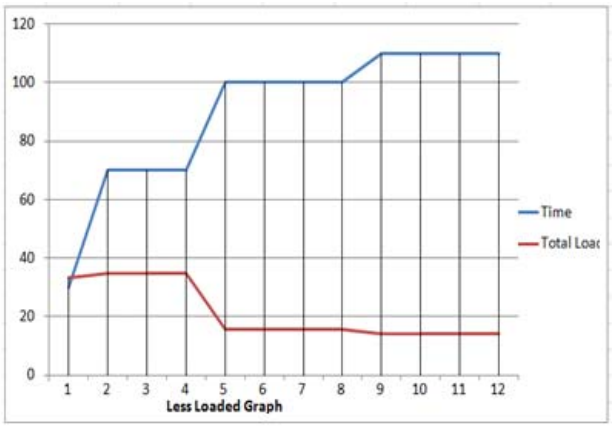

Figure 5: Less Loaded Graph

If Idle queue is empty than only less loaded queue will be selected. If the less loaded queue is not empty randomly VM will be selected

3.3. Scenario 3 : Medium Loaded Condition

Load is greater than 35 or less than equal to 70. We send cloudlet to the different VM according to that we took reading about the present status of the VM and if VM total load calculated based on (Network, Processor and Ram) is greater than 35 (Thirty five) or less than equal to 70 (Seventy) the status table will mark it as Less Loaded VM. We generate the graph based time versus load parameter on the different situation. Figure 6 shows the graph with Less Loaded condition

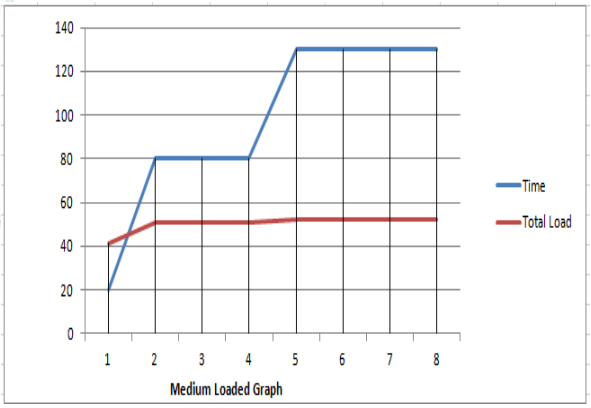

Figure 6: Medium Loaded Graph

If idle and Less Loaded VM queue is empty than task will select the VM from the medium loaded queue

3.4. Scenario 4 : heavily Loaded Condition

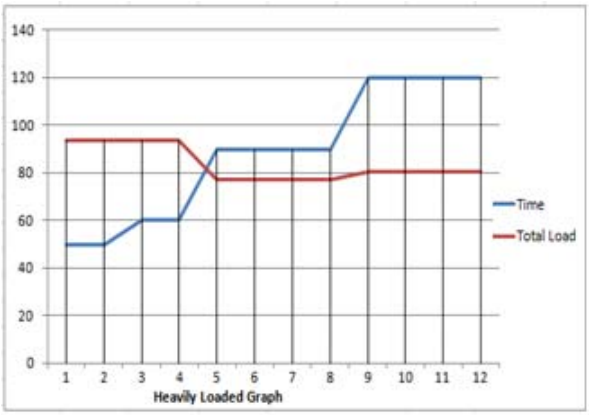

Figure 7: Heavily Loaded Graph

Load is greater than 70. We send cloudlet to the different VM according to that we took reading about the present status of the VM and if VM total load calculated based on (Network, Processor and Ram) is greater than 70 (Seventy) the status table will mark it as Less Loaded VM. We generate the graph based time versus load parameter on the different situation. Figure 7 shows the graph with Less Loaded condition. If idle, Less Loaded, Medium Loaded VM queue is empty than task will select the VM from the heavily loaded queue.

4. Conclusion

We already know that Load Balancing means every node in the network has exactly the same load. With the help of frequency based distribution of we divide total load in to the

four parts and generates flags according to present condition of VM. The four scenario shows that at particular time frame between 0to 0.6 most of the nodes are in the idle condition. Between 15 to 35 VMs are in the less loaded condition. Between 40 to 55 VMs are in the Medium Loaded Condition. And between 75 to 95 VMs are in the heavily loaded condition. With four different scenarios it is clear that each and every node has nearly the same load.

References

[1] NIST Cloud Definition, http://www.nist.gov/itl/csd/cloud-102511.cfm
[2] Gartner IaaS Definition, http://www.gartner.com/it-glossary/infrastructure-as-a-service-iaas, 2014
[3] Gartner PaaS Definition, http://www.gartner.com/it-glossary/platform-as-a-service-paas, 2014
[4] Gartner SaaS Definition, http://www.gartner.com/it-glossary/software-as-a-service-saas, 2014
[5] Microsoft Academic Research, Cloud computing, http://libra.msra.cn/Keyword/6051/cloud-computing?query= cloud%20computing, 2012.
[6] Google Trends, Cloud computing, http://www.google.com/trends/explore#q=cloud%20computing, 2014
[7] Saomya Ray, Ajanta De Sarkar, "Execution Analysis of Load Balancing Algorithm in cloud computing environment". International Journal on Cloud Computing : Services and Architecture (IJCCSA), Vol. 2, No. 5, October 2012
[8] Gowtha Kanagaraj, Naveen Shanmugasundaram And Satish Prakash, "Adaptive Load Balancing Algorithm Using Service Queue", 2nd International Conference on Computer Science and Information Technology (ICCSIT'2012) Singapore April 28-29, 2012.
[9] Jasmin James, "Efficient VM Load Balancing Algorithm for a Cloud Computing Environment", International Journal on Computer Science and Engineering, Vol. 4, September, 2012
[10] Tanveer AHmeda, Yogendra Singh, "Analytic Study of Load Balancing Techniques Using Tool Cloud Analyst:, International Journal of Engineering Research and Application (IJERA), Vol. 2, Issue 2, Mar – Apr 2012, pp. 1027-1030
[11] Gaochao Xu, Junjie Pang, and Xiaodong Fu, "A load balancing model based on cloud partition based on public cloud", IEEE, Volume 18, Number 1, February 2013
[12] Gowtham Kanagaraj, Naveen Shanmugasundaram And Sathish Prakash, "Adaptive Load Balancing Algorithm Using Service Queue", 2nd International Conference on Computer Science and Information Technology (ICCSIT'2012) Singapore April 28-29, 2012
[13] VIkas Patel, "Communication aware load balancing algorithm in cloud computing", International Journal of Computer Science and Management Research, Vol 2 Issue 5 May 2013
[14] Meenakshi Sharma, Pankaj Sharma, Dr. Sandeep Sharma, "Efficient Load Balancing Algorithm in VM Cloud Environment", IJCST Vol. 3, Issue 1, Jan. - March 2012

Author Profile

Hiren Parmar received the B.E. degree in Computer Engineering from Shantilal Shah Engineering College in 2011. Now he is studying his Master of Engineering from L. D. Collage of Engineering since 2012. Now he is doing his dissertation in Load Balancing on cloud.

Prof. Tushar Champaneria received the B.E degree in Computer Engineering from Saurshstra University in 2006 and M.E. degrees in Computer Science from Bira Institute of Technology and science in 2008. Presently he is working as Assistant Professor at L.D. College of Engineering.