ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly Peer Reviewed Journal)

Vol. 5, Issue 1, January 2016

Dynamic Queue Based Enhanced HTV Dynamic Load Balancing Algorithm in Cloud Computing

Divya Garg[1], Urvashi Saxena[2]

M.Tech (ST), Dept. of C.S.E, JSS Academy of Technical Education, Noida, U.P.,India[1]

Assistant Professor, Dept. of C.S.E., JSS Academy of Technical Education, Noida, U.P.,India[2]

ABSTRACT- Cloud computing is a very rapidly growing technology now a days. Almost every organization using Cloud computing because of their advantages. Cloud computing is an Internet –based development in which virtual resources are provided as a service over the internet .The more the number of users on the cloud ,the greater will be the load. Cloud service providers are required to balance this load effectively and efficiently so as not to degrade the performance. So load balancing is a very important issue which aims at distributing the load across multiple machines in an even and fair manner so that no single node is overwhelmed. In this paper, we aim to manage the distribution of resources in such a manner so as to avoid system bottlenecks. This paper will be presenting the issues of existing load balancing algorithms and also presenting the algorithm to overcome those issues.

KEYWORDS: Cloud computing, Load balancing, Virtual machine, Resource allocation.

I. INTRODUCTION

Cloud Computing is the latest developments of computing models after distributed computing, parallel processing and grid computing. Cloud Computing is a catagory of distributed computing where Different massively scalable IT related resources are provided to a number of external users as a service using internet. It is an on demand service in which shared resources, information, software and other devices are provided according to the clients requirement at specific time.

The main feature of Cloud Computing is that it makes all the resources available at one place in the form of a cluster and the resources are allocated to the users according to their requests. This cluster based approach helps in achieving the maximum CPU utilization and reduces the effort of users to access the cloud resources.

The main part of Cloud Computing is so called "cloud". Cloud is a group of computers – personal computers or servers- which are interconnected together. Cloud is the network that provides resources to the clients.

In 2011, NIST defined Cloud Computing as a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable (e.g. , networks, applications, storage, services and servers) that can be rapidly provisioned and released with minimal management effort or service provider interaction. According to **NIST** there are three service models, four deployment models and five characteristics.

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly Peer Reviewed Journal)

Vol. 5, Issue 1, January 2016

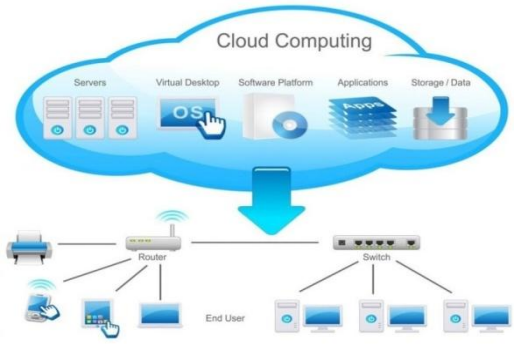

Figure 1- Cloud Computing Layout Diagram

Figure 2 shows the Services of cloud computing which is divided into Infrastructure-as-a-service (IaaS), Platform-as-a-service (PaaS) and Software-as-a-service (SaaS).

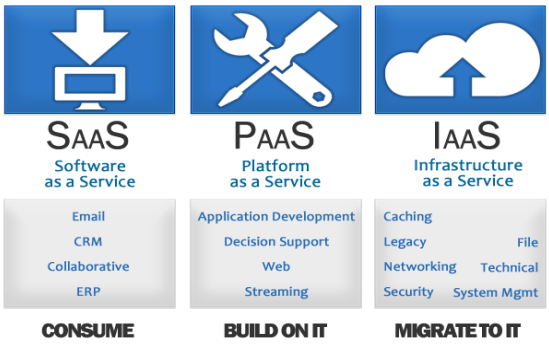

Figure 2 – Service Models

Figure 3 shows Four deployment model in cloud computing includes- Public Cloud, Private Cloud, Hybrid Cloud and Community Cloud.

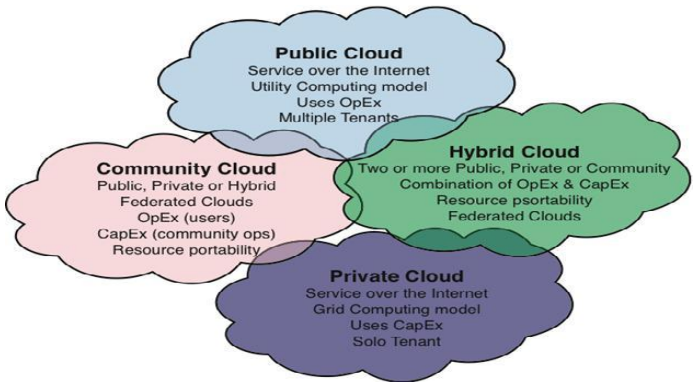

Figure 3 – Deployment Models

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly Peer Reviewed Journal)

Vol. 5, Issue 1, January 2016

The Five important characteristics of Cloud Computing are-

1. **On-Demand self service**: In this a user can use the computing capabilities on-demand automatically without requiring any interaction from service provider.

2. Rapid Elasticity: It provides resources and services which can be rapidly and elastically provisioned.

3. Broad Network Access: Where the resources and services are available over the network and can be accessed using heterogeneous thick and thin client (Laptops, PDAs, Mobile phones).

4. Resource Pooling: It provides a pool of resources which can be accessed using a multi-tenant model.

5. Measured Services: It works on pay-as-per use model.

II. LOAD BALANCING

Load Balancing is one of the central issues in Cloud Computing. The load can be CPU Load, memory capacity, delay and network load. The distribution of loads to the processing elements is known as load balancing. The goal of Load balancing algorithms is to maintain the load to each processing element such that all the processing elements become neither overloaded nor idle. By load balancing strategy it is possible to make every processor equally busy and to finish the works approximately at the same time. The aim of load balancing algorithm is dynamic in nature which does not consider the previous state .

Goals of Load balancing are:

- To improve the performance substantially,
- To have a backup plan in case the system fails even partially,
- To maintain the system stability,
- To accommodate future modification in the system.

Types of Load balancing algorithms

Depending on who initiated the process, load balancing algorithms can be of three catagories –

1. **Sender Initiated**: If the load balancing algorithm is initialised by the sender.
2. **Receiver Initiated**: If the load balancing algorithm is initiated by the receiver.
3. **Symmetric**: It is the combination of both sender initiated and receiver initiated.

Depending on the current state of the system, load balancing algorithms can be divided into 2 catagories –

1. **Static**: It doesnt depend on the current state of the system. Prior knowledge of the system is needed.

2. **Dynamic**: Decisions on load balancing are based on current state of the system. No prior knowledge is needed. So it is better than static approach.

III. RELATED WORK

Now a day's every organization are propagating towards the use of cloud computing. With no doubt we can put it as within few years, there will be lots of user for cloud computing. During that period of time, cloud provider will need to maintain more effective load balancing and an efficient resource management than the current scenario.
There are several static and dynamic type of load balancing algorithms on which various researches have been made. Static scheduling are suitable for small distributed environments with high Internet speed ignoring the communication delay. While algorithm which is based on APN takes communication delay and execution time into consideration and

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly Peer Reviewed Journal)

Vol. 5, Issue 1, January 2016

are suitable for larger distributed environments. In dynamic scheduling algorithm, some algorithms guarantee the load balancing and load sharing through self-adapting and intelligent distribution. In mixed scheduling algorithm, it mainly emphasizes on equal distribution of assigned computing task by reducing the communication cost of distributed computing nodes and at the same time it realizes balanced scheduling according to the computing volume of every node. Researchers have also conducted studies on algorithms of autonomic scheduling, central scheduling, intelligent scheduling and agent negotiated scheduling.

The other load balancing algorithm like random scheduling method randomly distributes load across the available servers, picking up one via random number generation and sending the current connection to it. While it is available on many load balancing products, its utility is questionable except where uptime is concerned – and then only we can detect down machines. The problem with this strategy is the load that is distributed among the nodes but there is no guaranties that load will be distributed equally.

IV. LOAD BALANCING ALGORITHM

Some of the existing load balancing algorithms has been discussed here:

1. **ROUND ROBIN ALGORITHM**- In Round Robin (RR), it passes each new connection request to the next server in queue, eventually distributing connections evenly across the array of machines being load balanced. The basic purpose of working behind the algorithm is the system builds a standard circular queue and walks through it, sending one request to each machine before getting to the start of the queue and repeating it again.

2. **DYNAMIC ROUND-ROBIN ALGORITHM-** In this dynamic load balancing strategy, distributions of connections is done on the basis of server performance analysis such as the current number of instances i.e. connection per node or the response time of a fastest node. This type of an application level connection distribution controlling method is rarely available in a conventional load balancer.

3. **ADAPTIVE RESOURCE ALLOCATION ALGORITHM (ARA)**- In ARA algorithm for adaptive resource allocation in cloud systems, which attempts to counteract the deleterious effect of burrstones by allowing some randomness in the decision making process and thus improve overall system performance and availability.

4. **EQUALLY SPREAD CURRENT EXECUTION ALGORITHM-** In this algorithm, processes are handled with priorities. It distribute the load randomly by checking the size and transfer the load to that virtual machine which is lightly loaded or handle that task easily and take less time by giving maximum throughput. It is spread spectrum technique.

5. **THROTTLED ALGORITHM**- This algorithm is completely based on virtual machine in which, the client first requests the load balancer to find a suitable Virtual Machine (VM) which access that loads easily and perform the required operation.

V. PROBLEM DESCRIPTION

As we shown above there are many algorithms available for load balancing . when we campare these algorithms, there are several disadvantages. The overall response time of Round Robin policy and Equally Spread Current Execution policy is almost same while that of Throttled policy is low as compared to other two policies.

Issue in ROUND ROBIN strategy is that it will not check whether the server is heavy loaded or not, it will directly assign the request whenever its turn comes. Because of this some server are heavy loaded while some are lightly loaded.

Disadvantage of ADAPTIVE RESOURCE ALLOCATION ALGORITHM is that it considers the Poisson arrival streams as well as the exponentially distributed service time and the fixed number of choice.

The problem in THROTTLED ALGORITHM is that, it does not consider the advanced load balancing requirements such as processing times for each individual requests and it does not measure the performance of data center and the load on the data center.

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly Peer Reviewed Journal)

Vol. 5, Issue 1, January 2016

Although there are many algorithms available for load balancing in Cloud computing but on analyzing them thoroughly we still feel the need of further research so as to improve the performance and efficiency of the algorithms. The current load balancing algorithm distributes the workload among all the nodes in a round robin fashion i.e. it allocates the first request to the very first node in the queue, then the second request is allocated to the second node in the queue if enough resources are available on that node otherwise it moves to the next node in the queue, and when the end of the queue is reached it will again start from the very first node in the queue. Here the problem is that there is no resource monitoring hence it does not have any idea about the node whom it assigns the request is heavily loaded or lightly loaded. So some nodes may be heavily loaded while others are idle but still round robin will assign the request to that heavily loaded node if it is the next node in the queue which will degrade the performance and efficiency of that heavily loaded node. So to overcome this problem a new algorithm (**HTV Dynamic Load Balancing Algorithm**) was proposed in which continuous monitoring of the resources are done to know the status of each and every node and queue is maintained in which the weight factor will be stored and update whenever continuous monitoring is done. When request comes, the resources will be allocated from the information present in the queue dynamically to balance the load on nodes. But still there is some problem which needs to be worked out. The problem of time limit of the requests (tasks) that needs to be executed. Some tasks have lower time limit that the other ones. Such task needs to be serviced prior than the other ones.

VI. PROPOSED ALGORITHM

We are proposing an algorithm in which there will be a continuous monitoring of the resources so as to know the status of available resources on each node as well as there will also be a concept of time limit associated with each task of the incoming tasks or requests that needs to be serviced.

PSEUDO CODE

Step 1: [Calculate the Load, L];

L <- (Total _Resources - Used _Resources);

// where L is free memory in terms of percentage.

Step 2: [Calculate Performance Factor P];

P1 <- average (current_response_time)

P <- P1 – (previously calculated Y1)

P <- P/(previous P1) * 100 // counting P in terms of previously counted P1.

Step 3: [Finding W];

W <- L – P:

If (W< 0)

W = 0;

Step 4: [find minimum of all W expect the nodes with W value 0]

Min_W= min (all W's)

Step 5: [Find Min_factor and divide all W by that factor]

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly Peer Reviewed Journal)

Vol. 5, Issue 1, January 2016

Min_factor <- Min_W

W <- W/Min_factor

Step 6: [Generate Dynamic Queue on base of W]

Step 7: Calculate stand-by time (ST) for all tasks

[ST = time limit (TL) – time required for execution];

Step 8: Arrange all the tasks in ascending order of their stand-by time (ST).

Step 9: Allocate a task with minimum time limit (TL) to the virtual machine having maximum performance factors.

EXPLANATION

Step-1: The value of load factor L is calculated by using the fomula: L = (Total_Resources – Used_Resources) , by this we get the available free load on nodes.

Step-2: To get an idea about the increase or decrease of performance , the performance factor P is calculated by sending a request at regular interval to all nodes and the response time is calculated(request time +response time). Every time the value of P will be different and averaging them P1 will be calculated , and then the previously calculated P1 will be subtracted from current values P1 to get the performance .

Step-3: Calculate W (mathematical function to count parameter value for each node) using formula : W = L-P , here we are interested in node with lowest response time hence we subtract P from L.

Step-4: Once W is calculated then the minimum of all W which we calculated is stored in Min_W.

Step-5: After getting the Min_factor , divide all the W values by that factor .

Step-6: On the basis of this W a queue is generated.

Step 7- Calculate minimum stand-by time (ST) for each task , using the formula

Step 8- Algorithm is progressed step by step arranging all the incoming tasks in the ascending order of their time limit so as to make sure that the tasks with shorter time limit will be serviced before the tasks with longer time limit otherwise the tasks with shorter time limit would have been dropped .

Step 9- The sorted tasks will then be allocated to suitable VMs according to the information of VMs present in the dynamic queues.

VII. CONCLUSION AND FUTURE WORK

This paper is proposing an **enhanced priority based HTV load balancing algorithm** to perform the effective and reliable resource allocation of the tasks on the servers in cloud computing environment. The proposed algorithm considers the three parameters load on the server , current performance of server and time limit of the tasks as its prime parameters. In this, the proposed algorithm is calculating the load and performance factor of each virtual machine and then allocating the incoming tasks to various virtual machines according to their time limit and stand-by time. According to the proposed algorithm we conclude that by considering the time limit of the tasks, our algorithm will increase the throughput and performance.
In this work some parameters are still left to be incorporated. So, in future, we plan to find out the other technique of load balancing in cloud computing which still needs issues to be addressed. Future work includes the implementation of

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly Peer Reviewed Journal)

Vol. 5, Issue 1, January 2016

the proposed work in the existing Cloud computing architecture for effective utilization of resources and better performance.

REFERENCES

1. Jitendra Bhatia, Tirth Patel, Harshal Trivedi, Vishru Majmudar "HTV Dynamic Load Balancing Algorithm for Virtual Machine Instances in Cloud", IEEE Computer Society, 2012 International Symposium on Cloud and Services Computing.
2. Klaithem Al Nuaimi, Nader Mohamed, Mariam Al Nuaimi and Jameela Al-Jaroodi "A Survey of Load Balancing in Cloud Computing: Challenges and Algorithms" 2012 IEEE Second Symposium on Network Cloud Computing and Applications
3. Martin Randles, David Lamb, A. Taleb-Bendiab "A Comparative Study into Distributed Load Balancing Algorithms for Cloud Computing", 2010 IEEE 24th International Conference on Advanced Information Networking and Applications Workshops
4. Namrata Swarnkar, Asst. Prof. Atesh Kumar Singh, Dr. R. Shankar "A Survey of Load Balancing Techniques in Cloud Computing", International Journal of Engineering Research & Technology (IJERT)Vol. 2 Issue 8, August – 2013
5. Tejinder Sharma, Dr.Vijay Kumar Banga, " Proposed Efficient and Enhanced Algorithm in Cloud Computing", International Journal of Engineering Research & Technology (IJERT) Vol. 2 Issue 2, February- 2013 ISSN: 2278-0181
6. Wei Zhao, Yong Peng, Feng Xie, Zhonghua Dai "Modeling and Simulation of Cloud Computing: A Review", 2012 IEEE Asia Pacific Cloud Computing Congress (APCloudCC)
7. Ling Leng, Lin Wang "Research on cloud computing and key technologies", 2012 International Conference on Computer Science and Information Processing (CSIP), 2012 IEEE
8. Khiyaita, M. Zbakh, H. El Bakkali, Dafir El Kettani "Load Balancing Cloud Computing", State of Art: 2012 IEEE