

# A Novel Approach for Load Balancing in Cloud Environment

Megha Dabas<sup>1</sup> & Pooja<sup>2</sup>

<sup>1</sup>Department of Computer Science, International Institute of Technology and Business (IITB), Deenbandhu Chhotu Ram University of Science & Technology (DCRUST), Sonapat

<sup>2</sup>Department of Computer Science, International Institute of Technology and Business (IITB), Deenbandhu Chhotu Ram University of Science & Technology (DCRUST), Sonapat

Received: December 06, 2018

Accepted: January 07, 2019

**ABSTRACT:** Cloud computing environment provides multiple resources and services for sharing with their clients. Data storage on cloud computing is increasing day by day which causes scarcity of resources on cloud data centers. Also some data centers are overloaded & some are under loaded. Therefore load balancing on cloud data centers is required. With load balancing concept some tasks of overloaded servers are transferred to under loaded servers. Generally load balancing algorithms work dynamically. There are multiple dynamic load balancing algorithms exists for balancing the work load on cloud data centers. In this paper new approach is applied for load balancing in cloud computing environment.

**Key Words:** Cloud Computing, Load Balancing, First Come First Serve, Throttle Method, Equally Spaced

## I. INTRODUCTION

Data storage on cloud computing is increasing day by day which causes scarcity of resources on cloud data centers. Cloud computing provides flexible way to retain data and files which involves virtualization, distributed computing, and web services. The main aim of cloud servers is to share vast amount of resources with their clients with low cost. The clients can use the cloud resources by registering with specific server & send requests for the resources. The server after authentication provides desired services to the requesting clients. The cloud computing now a day is facing a real time challenge of load balancing [1]. The main reason for this challenge is the increase in the users demand for cloud services. So it is practically impossible to maintain the one or more free service to fulfill the demand. Providing each server with one demand to fulfill will result into traffic on the server and ultimately the crash of the system. It is used by Cloud service provider (CSP) in its own cloud computing platform to provide a high efficient solution for the user. Also, a inter CSP load balancing mechanism is needed to construct a low cost and infinite resource pool for the consumer [2].

A general model of Cloud Computing is shown in figure 1 below.

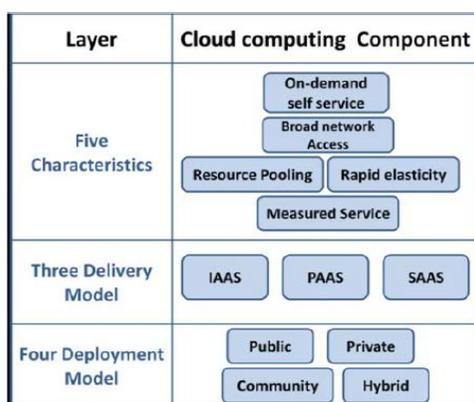


Figure 1: Model of Cloud Computing

Load balancing is helped to distribute the dynamic workload across multiple nodes to ensure that no single node is overloaded. It helps in proper utilization of resources. It also improves the performance of the system. Many existing algorithms provide load balancing and better resource utilization[3]. Load balancing is the process of finding overloaded nodes and then transferring the extra load to other nodes. In this paper Hybrid approach is applied for load balancing using Round Robin, Equally Spread Current Execution and (ESCE) Throttled algorithms.

## II. LOAD BALANCING IN CLOUD COMPUTING

Load balancing is used to distributing a larger processing load to smaller processing nodes for enhancing the overall performance of system. In cloud computing environment load balancing is required distribute the dynamic local is a techniques that helped networks and resources by providing a Maximum throughput with minimum

response time. Load balancing is dividing the traffic between all servers, so data can be sent and received without any delay with load balancing. Load balancing is required to improve the performance of the system by minimize the overall completion time and avoid the situation where some resources are heavily loaded or others remains under loaded in the system [4].

The goals of load balancing are:

- Improve the performance
- Maintain system stability
- Build fault tolerance system
- Accommodate future modification.
- Energy is saved in case of low load
- Maximize throughput of the system
- Minimize communication overhead
- Resources are easily available on demand
- Resources are efficiently utilized under condition of high/low load
- Minimize overall completion time (makes pan)

Figure 2 below shows load balancing in cloud environment.

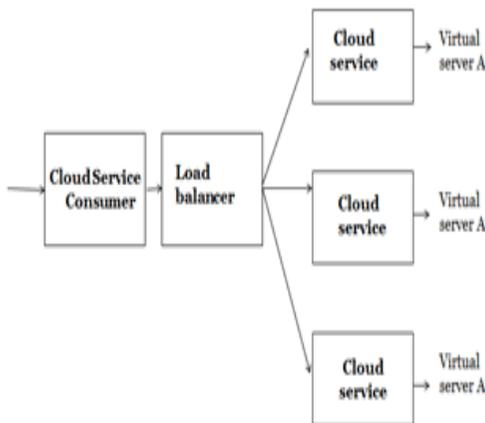


Figure 2: Load Balancer in cloud Environment

### III. EXISTING LOAD BALANCING ALGORITHMS

To design an effective load balancing policy and to determine how to increase the cloud resource usage are the two main goals of a cloud service provider. The VM scheduling algorithms for load balancing helps in allotment of VMs efficiently on need. Basically, a VM load balancing algorithm decides which VM is to allocate when request is made by cloud consumer. Numerous VM load balancing algorithms that have been proposed are discussed here [6][10]:

#### 1. Round Robin Load Balancing Algorithm

It is a very simple load balancing algorithm that places the newly coming request on the available virtual machines in a circular manner. The major advantage of this algorithm is its simplicity and easy implementation. The main drawbacks are that it requires the prior knowledge of user tasks and system resources & it do not make use of current state of the system

#### 2. ESCE load Balancing Algorithm

ESCE stands for Equally Spread Current Execution. It is also called Active VM Load Balancing algorithm. This algorithm is based on spread spectrum technique. As the name indicates, it equally distributes the workload on each VM in data center. A job queue keeps all the cloudlet requests that need the VM for their execution. ESCE VM Load Balancer (VMLB) also maintains a list of virtual machines. The VM Load Balancer continuously checks the job queue and VM list. If a VM is found free, then cloudlet request will be allotted over that VM. At the same time, VMLB inspect the overloaded VMs. If any virtual machine is found overloaded, then VMLB move some load to an idle or an under loaded virtual machine, so as to reduce some load of overloaded VM. The main drawback is high computational overhead.

#### 3. Throttled Load Balancing Algorithm

It is a dynamic approach. In this, user submits its request to the Data Center Controller(DCC). Data Center Controller asks the VM Load Balancer to determine the appropriate virtual machine that can handle that much workload easily. Throttled VM Load Balancer keeps a virtual machine list and their status (available/busy). If a suitable VM is found on memory space, cores or availability basis, then throttled VM Load Balancer accept the cloudlet request and allot the cloudlet request over that virtual machine. Otherwise, clients have to wait in the waiting queue until a suitable VM becomes available. Among all, it is best approach for load balancing, since it maintains the present state of all VMs in data center. But the major drawback is that it works properly only if all VMs in a data center have same hardware configuration.

### IV. PROPOSED WORK

The goal of the proposed work is to design an efficient scheduling algorithm that uniformly distribute workload among the available virtual machines in a data center and at the same time, decrease the overall response time and data center processing time. The proposed approach is a combination of Throttled (TVLB) and ESCE

algorithm. TVLB algorithm makes use of states of VMs. A virtual machine state may be either AVAILABLE or BUSY. AVAILABLE state indicates that the virtual machine is idle/free and ready for cloudlet allotment, where BUSY state indicates that the current virtual machine is busy in execution of previous cloudlets and is not available to handle any new cloudlet request. This current load state of a VM helps in taking decision whether to allocate cloudlets to virtual machines or not.

Active VM Load Balancing algorithms continuously monitor the job queue for new cloudlets and allot them to the bunch of idle/free VMs. It also maintains the list of cloudlets allocated to each virtual machine. This allocated cloudlet list helps in determining whether a VM is overloaded or under loaded at particular moment of time. On the basis of this information, VM load Balancer moves some load from overloaded VMs to the VM having minimum number of cloudlets, so as to maintain a high degree of balance among virtual machines. These features combined together, make the proposed scheduling algorithm more efficient & effective and help in fair distribution of the load.

Figure 3 below shows the proposed load balancing algorithm.

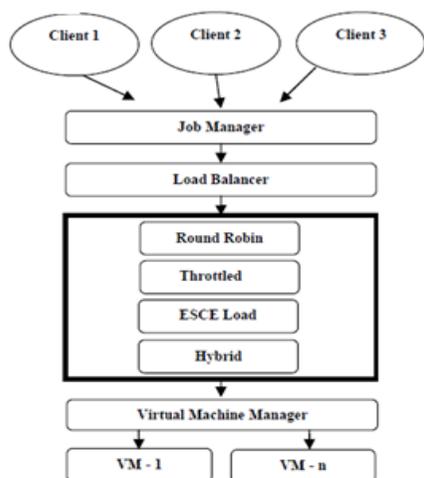


Figure 3: load balancing Algorithms

### V. IMPLEMENTATION & RESULTS

Our implementation for performing load balancing on cloud environment using hybrid approach of first come first serve, throttle and equally spaced algorithm work as follows:

Initially some servers on cloud data centers startup & wait for user’s requests. As user’s requests arrive, they are performed in round robin fashion i.e., as soon as client’s request arises and server status is available then the server is allocated to the requesting client. Now after

allocating server the throttle load balance technique checks the requested resources (Number of CPUs, Memory, Processing time etc.) & allocate the cloud data center server to the client that satisfying the client need. The process also checks the desired threshold value for each server. If the server utilization exceeds the specified threshold (>75%) then the server of another cloud data center will start for satisfying the request.

When the client stops the task then the service allocated to the client is released & same can be reallocated to another client in the waiting. Now our implementation performs equally spread current execution algorithm to equally distribute the load on various active cloud data center servers.

We have use another threshold value that becomes the basic of balancing the load on servers. If load of any active server is less than specified threshold value (<25%) and there is another server with required space then load of first server is transferred to second server and first server is closed. Hence the work provides efficient load balancing among the active servers and server with low utilization is closed after moving their load to another server with required space.

Figure 4 shows the main menu of our implementation.

```

Output - Load_Balance (run)
run:
Select Your choice
For Allocation PRESS : 1
For Deallocation PRESS : 2
For Load Balance PRESS : 3
For Exit PRESS : 4
Enter Your Choice:
  
```

Figure 4: Main menu of implementation

When user select option 1 i.e., allocation of resources then it ask for client id and numbers of CPUs required as shown in figure 5.

Figure 5 shows the effect after allocation of resources.

```

Output - Load_Balance (run)
Enter the type of VM Required :
4
scheduler is working with cpu req = 16 for the client id 1
New utilization of server with ID 101 is 50.0
Server id 101 is allocated to client id 1
***** Now remaining CPU of server 101 is 16 utilization = 50.0*****

Server Status
ServerId    RemainingCap  Utilization
101         16           50.0
102         32           0.0

Want to continue y/n
  
```

Figure 5: Effect after resource allocation

When we done with allocation then press n. Main menu options will display again on the screen as shown in figure 6.

```

Output - Load_Balance (run) %
>> Want to continue y/n
>> n
>> Select Your choice
>> For Allocation PRESS : 1
>> For Deallocation PRESS : 2
>> For Load Balance PRESS : 3
>> For Exit PRESS : 4
>> Enter Your Choice:
2|
    
```

Figure 6: Selecting de allocation choice

In main menu options now select the option 2 for deallocation of resources. It will ask for client id and numbers of CPUs to be deallocated. If client id is not correct or numbers CPUs to be deallocated are more than the allocation then appropriate error message is displayed. If everything is OK then deallocation is performed. The remaining capacity and utilization value are updated for the server on which deallocation are performed as shown in figure 7.

```

Output - Load_Balance (run) %
>> From which client you want to deallocate
>> 2
>> No. of CPU You want to deallocate :
>> 4
>> Deallocation of 4 no of cpu is performed from clien id 2
>> Server on which deallocation is performed is 101
>> Remaining capacity of cpu is 16
>> Now the server utilization is 50.0
    
```

Figure 7: Effect of deallocation

Now press the option 3 for performing the load balance as shown in figure 8 below.

```

Output - Load_Balance (run) %
>> Select Your choice
>> For Allocation PRESS : 1
>> For Deallocation PRESS : 2
>> For Load Balance PRESS : 3
>> For Exit PRESS : 4
>> Enter Your Choice:
3|
    
```

Figure 8: Selecting the option 3 for load balancing

Figure 9 below shows the effect of load balancing.

```

Output - Load_Balance (run) %
>> server id 103 is under utilize with utilization value of 25.0
>> Server id 101 is the nearest server for moving of resource
>> Resource of server id 103 is moving to server id 101
>> Resource of server id 103 is transferred to server id 101

Server Status

ServerID      Utilization
101           75.0
102           75.0
103           0.0
    
```

Figure 9: Effect of load balance

In this work, we propose the hybrid approach of load balancing algorithms to overcome the drawback of existing methods. We first assign the load on the server using first come first serve fashion then we use throttle concept to find the suitable machine for current tasks and finally we use equally spread technique to equally distribute the load on various virtual machines to balance the load [11][12].

**REFERENCES**

1. Vishwas Bagwaiya, Sandeep k. Raghuwanshi, "Hybrid Approach Using Throttled And ESCE Load Balancing Algorithms In Cloud Computing".
2. Mamta Khanchi, Sanjay Tyagi, "AN EFFICIENT ALGORITHM FOR LOAD BALANCING IN CLOUD COMPUTING",© International Journal of Engineering Sciences & Research Technology.
3. Bhavisha Patel, Shreyas Patel, "Various Load Balancing Algorithms in cloud Computing",IJARIE-ISSN(O)-2395-4396. Vol-1 Issue-2 2015.
4. MithunDsouza, Mohammed Rizwan, RamnathGaonkarand, S. Sathyanarayana, "SCHEDULING AND LOAD BALANCINGTECHNIQUES IN CLOUD COMPUTING: ASURVEY",International Journal of Latest Trends in Engineering and Technology Special Issue SACAIM 2016, pp. 309-316 e-ISSN:2278-621X.
5. Bhasker Prasad Rimal, Enumi Choi, "A Taxonomyand survey of cloud computing systems", 2009 Fifth International Joint Conference on INC, IMS and IDC.
6. Rajwinder Kaur and Pawan Luthra, "Load Balancing in Cloud Computing".
7. Amandeep Kaur Sidhu, Supriya Kinger,"Analysis of Load Balancing Techniques in Cloud Computing", International Journal of Computers &Technology Volume 4 No. 2, March-April, 2013, ISSN 2277-3061.
8. Ratan Mishraand, Anant Jaiswa, "Ant colony Optimization: A Solution of Load balancing in

- Cloud”, International Journal of Web & Semantic Technology (IJWesT) Vol.3, No.2, April 2012
9. Fang, Y., Wang, F., and Ge, J., “A Task Scheduling Algorithm Based on Load Balancing in Cloud Computing”, Lecture Notes in Computer Science, Jg. 2010(6318): pp. 271-277, (2010).
  10. Abhinav Hans, Sheetal Kalra, “Comparative Study of Different Cloud Computing Load Balancing Techniques”, 2014 International Conference on Medical Imaging, m-Health and Emerging Communication Systems (MedCom).
  11. V RaviTeja Kanakala, V. Krishna Reddy and K. Karthik, “Performance Analysis of Load Balancing Techniques in Cloud Computing Environment” 978-1-4799-6085-9/15/\$31.00 ©2015 IEEE.
  12. Subhadra Bose Shaw, “A Survey on Scheduling and Load Balancing Techniques in Cloud Computing Environment”, 2014 5th International Conference on Computer and Communication Technology (ICCT).