



Improved Joint Ideal Queue (JIQ) Load Balancing Algorithm in Cloud Computing

¹Dr. Gaurav Sharma, ²Er. Manisha Gupta

¹Associate Professor, ²M.tech Student

^{1,2}Department of computer Science & Engineering JMIT, Radaur, Haryana, India

Abstract: *Cloud Computing is mostly describe as "cloud" is the delivery of demand services resources everything from application to data centers with the help of internet on a pay for use basis. In cloud computing Load Balancing is a scheme to distribute the workload for balancing between two or more cloud server. Load balancing scheme is define as some cases which are defining as guarantee continuity of the service provided manage, heavy traffic levels and always be ready for immediate peaks in traffics. Load balancing goal is to optimize resource usage, maintain the cost of data center and virtual machine, increase throughput, minimize response Time and avoid overload of any single Resource. The main aim of research paper is to reduce the VMcost , response times and DCprocessing time using Joint Idle queue and shortest queue across VM's in multi data center and optimize response time server broker policy. This paper also provide the result with the implementation of proposed algorithm ie.(Modified Joint Ideal queue).*

Keywords: *Cloud computing, load balancing, cloud Analyst, response time, Virtual Machine, data center, cost.*

I. INTRODUCTION

Cloud computing is simply describe as "cloud is delivery of demand services resources application to data centers over the internet on the pay or use basis. Cloud computing describe a new Class of network based on computing that takes place over the internet. In cloud computing main objective is to control the load over the network. This paper addressed the load balancing issue on the consumer side. All the load balancing method are designed to determine which VM is selected for the implementation of next task units so resource management play an important role in the performance of the whole cloud system and the level of user satisfaction provided by cloud system. Load balancing, as the name implies, is a scheme that allows workloads to be distributed across multiple resources, to make effective resource utilization and to give best response time by handling a condition in which some of the nodes are over loaded while some others are inferior loaded. The goal of load balancing is to optimize utilization and throughput but reducing the response time[1][2]. As on requirement self service model, load arrives in any way or dynamically in cloud computing environment which causes some VM servers to be highly loaded while others idle or lightly loaded which in turn leads to poor performance and make user unsatisfied[1]. There are several load balancing algorithm for the improvement and optimization of cloud performance parameter such as:

- A. **Make Span** :Total length of the schedule when all jobs are completed. This parameter should be minimum, because the lower time algorithm will take for completion good will be performance.
- B. **Overhead**: Measure the overhead needed for communication between the nodes due to movement of tasks.
- C. **Overall Performance**:Improved efficiency of the system at minimum cost with minimum response time.
- D. **Waiting time**:A time taken by a process in the queue to sit idle for achiving the chance for implementation after achiving resource or machine.
- E. **Cost** :Pricing of services on cloud.
- F. **Energy Consumption** :How much energy is generated by the nodes or data center.
- G. **Throughput** : The total no of task that have completed implementation is called throughput. A high throughput is needed for best performance of the system.
- H. **Associated Overhead** :The amount of work that is produced by the implementation of the LB algorithm. decrease overhead is common for successful implementation of the algorithm.
- I. **Fault Tolerant** :It is the ability to perform correctly and uniformly in the situation of failure at any arbitrary node in the system.
- J. **Migration Time** :The time is taken in migration or transfer of task from one machine to any other machine in the system. This time should be less for improving the performance of the System.
- K. **Response Time** :It is the minimum time that a distributed system implementing a specific load balancing algorithm takes to respond.
- L. **Resource Utilization** : It is the degree to which the resource of the system are utilized. A good load balancing algorithm provides maximum resource utilization.

- M. *Scalability*: It determine the ability of the system to accomplish load balancing algorithm with a restricted number of processors or machines.
- N. *Performance*: It represent the effectiveness of the system after performing the load balancing. If all the parameter are satisfied optimally then it will highly improve the performance of system.

II. RELATED WORK

- A. *Round Robin* – First request is allocated to a randomly selected VM. Subsequent requests are allocated in circular order. Advantages of this algorithm is Equal distribution of load occur. Disadvantage of this algorithm is Job processing time is not considered[4].
- B. *Weighted Round Robin* –Weight is allocated to each VM based on its processing capacity. More requests are allocated to the powerful VM. Advantage of this algorithm is Better resource utilization. Disadvantage of this algorithm is Processing time of each request is not considered[9].
- C. *Dynamic Round Robin*: This algorithm is based on two rules : i) Retiring State of VM and ii) Retirement Threshold and migration of VMs. advantage of this algorithm is Reduces the power consumption cost. Disadvantage of this algorithm is Does not scale up for large data centers[5].
- D. *Min-Min Scheduling Algorithm*- The smallest task is assigned to the fastest resource. The task is removed from the set and same process is repeated. Advantage of this algorithm is The method is simple. Disadvantage of this algorithm is Does not consider the existing load on a resource[6].
- E. *Max-Min Algorithm*: Works as the Min-Min algorithm. But jobs having large execution time are executed first. Advantage of this algorithm is Reduces the make span. Disadvantage of this algorithm is Smaller jobs have to wait for long time[8].
- F. *Throttled*- Record of the state of each VM (busy/idle) is maintained. Request is accepted if a match is found in the table otherwise -1 is returned and the request is queued. Advantage of this algorithm is TLB tries to distribute the load evenly among the VMs. Disadvantage of this algorithm is does not consider the current load on VM[3].
- G. *Modified Throttled*–Unlike Throttled here the index table is searched from the next to already assigned VM. Advantage of this algorithm is Gives better response time. Disadvantage of this algorithm is State of index table may change during next allocation[3].
- H. *Equally Spread Current Execution (ESCE) Algorithm* - Request is assigned to any available VM that can handle it[7]. If there is an overloaded VM then the balancer distributes some of the tasks to some idle VM to balance the load. Advantage of this algorithm is Response time and processing time of a job is improved. Disadvantage of this algorithm is Not fault tolerant because of single point of failure[5].

III. PROPOSED WORK

METHODOLOGY USED IN LOAD BALANCING SYSTEM

- The algorithm contains the primary and secondary load balancing systems, which communicate through a data structure called I-queue.
- I-queue is a list of a subset of processors that have reported to be idle. All processors are accessible from each of the dispatchers(d_i).
- Primary load balancer: Select the dispatcher(d_i) based on maximum queue length.

If the size of dispatcher(d_i) is greater than maximum queue length

```
if( $d_i.size() > max$ )
     $max = d_i.size();$ 
     $dis = d_i;$ 
```

then based on this (dis) object contain the selected dispatcher(d_i), selection of this IDEAL VM take place. If the selected virtual machine is null. Then we can select this virtual machine based on VMQUEUE.

```
for(Integer id :  $vmQueue.keySet()$ )
    If( $vmQueue.get(id) < min$ )
         $min = vmQueue.get(id);$ 
         $selVM2=id;$ 
         $selVM=selVM2;$ 
    return  $selVM;$ 
```

- Secondary load balancing. We consider two load balancing algorithms in the opposite direction: Random and SQ(d).

With JIQ-Random,: $if(secondaryPolicy.equals("Random"))$

```
Random  $rnd = new Random();$ 
int  $d = rnd.nextInt(numDispatcher);$ 
Dispatcher  $dis = disp.get(d);$ 
 $dis.addIdleVM(vmId);$ 
```

an idle processor select an I-queue uniformly at random, and with JIQ-SQ(d),
Dispatcher $sel=null;$

```

int min = 1000;
for(Dispatcher d : disp)
    if(d.size()<min)
        min=d.size();
    sel = d;
an idle processor select d random I-queues and joins the one with the smallest queue length.3.
    
```

IV. EXPERIMENTAL SETUP AND RESULTS:

In this the Experimental work is define by the cloud analyst simulator. Cloud analyst is based on the java programming language. And consist of a GUI interface which helps in easy configuration of attribute required for the experiments. It is a service model in which one or more key element of data element of data analytics is provided through a public and private cloud. Cloud simulator is used for managing the load over the network with the help of different load balancing policy. Cloud analyst define four main important simulation configure simulation, define internet characteristics, run simulation and show region boundaries. The region boundariesare(R0, R1, R2, R3, R4, R5) which define the six different region.

The purposed algorithm is implemented with the help of Cloud Analyst Simulator. Cloud analyst is Graphic user interface which is developed on cloud-sim architecture. In this Simulation parameter are User base configuration , data center configuration and advanced configuration.

In this paper the various Response time(millisecond) Graph and datacenter processing Time graph are shown below with their corresponding readings.

ALGORITHM		RESPONSE-TIME(m-s)	DC-PROCESSING -TIME(m-s)
JIQ	EXISTING	339.29	28.54
MODIFIED JIQ	PROPOSED	333.09	22.29

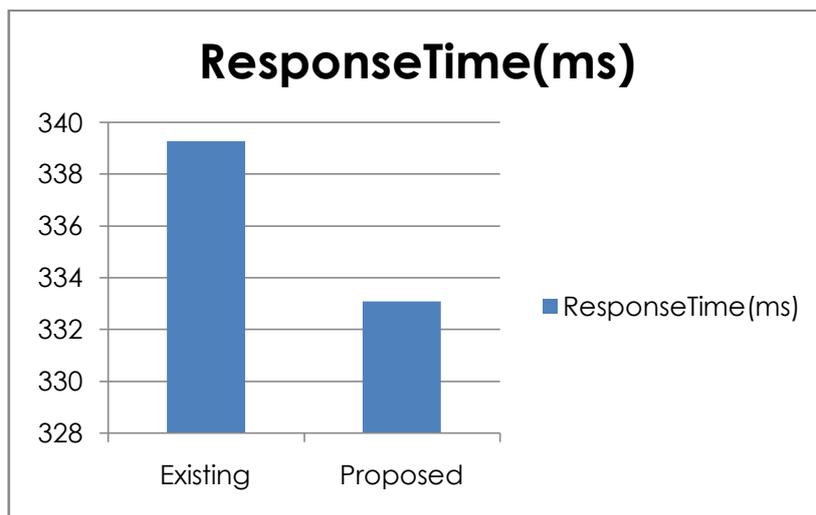


Figure1: Response Time (ms) of Existing and Proposed

This is the Response Time(ms) Graph of Existing and proposed Load balancing algorithm. In this Existing (JIQ) algorithm Response Time is 338.48 milliseconds. and proposed (Modified JIQ) algorithm Response Time is 333.33 milliseconds give better performance.

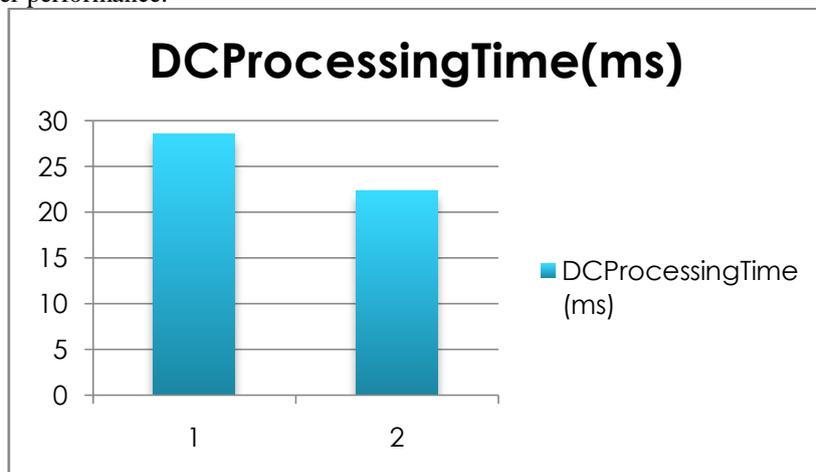


Figure 2: DCProcessing Time (ms) of Existing and Proposed

This is the graph of DC processing time(ms) of Existing and proposed load balancing algorithm. In this Existing (JIQ) algorithm DC processing Time is 28.43 and proposed (Modified JIQ) algorithm DC ProcessingTime is 22.53(ms).

VM	Existing (JIQ)	Proposed (Modified JIQ)
0	30082	30113
1	29715	30113
2	30103	30113
3	30615	30113
4	30051	30114

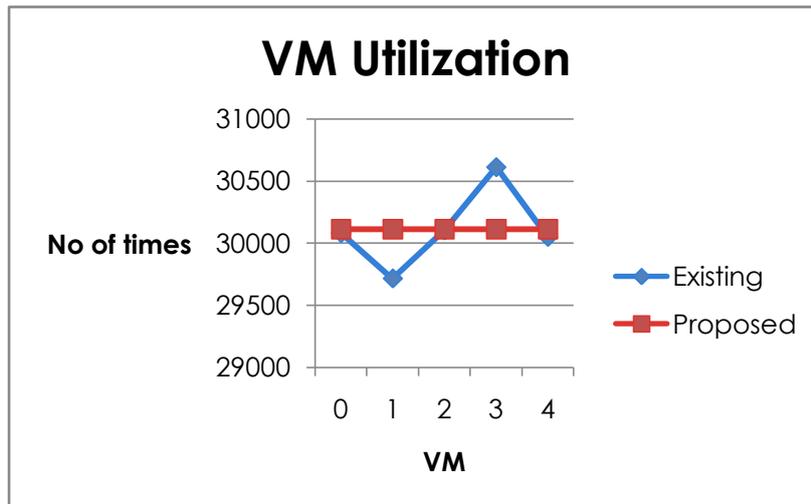


Figure 3: Response Time of Existing and Proposed

This is the virtual machine utilization graph in load balancing in cloud computing. In this Existing and proposed algorithm shows that number of time the virtual machine utilization take place.

The following given below snapshot of the JIQ Algorithm (Response time and datacenter processing time) and Modified JIQ Algorithm (Response time and datacenter processing time)

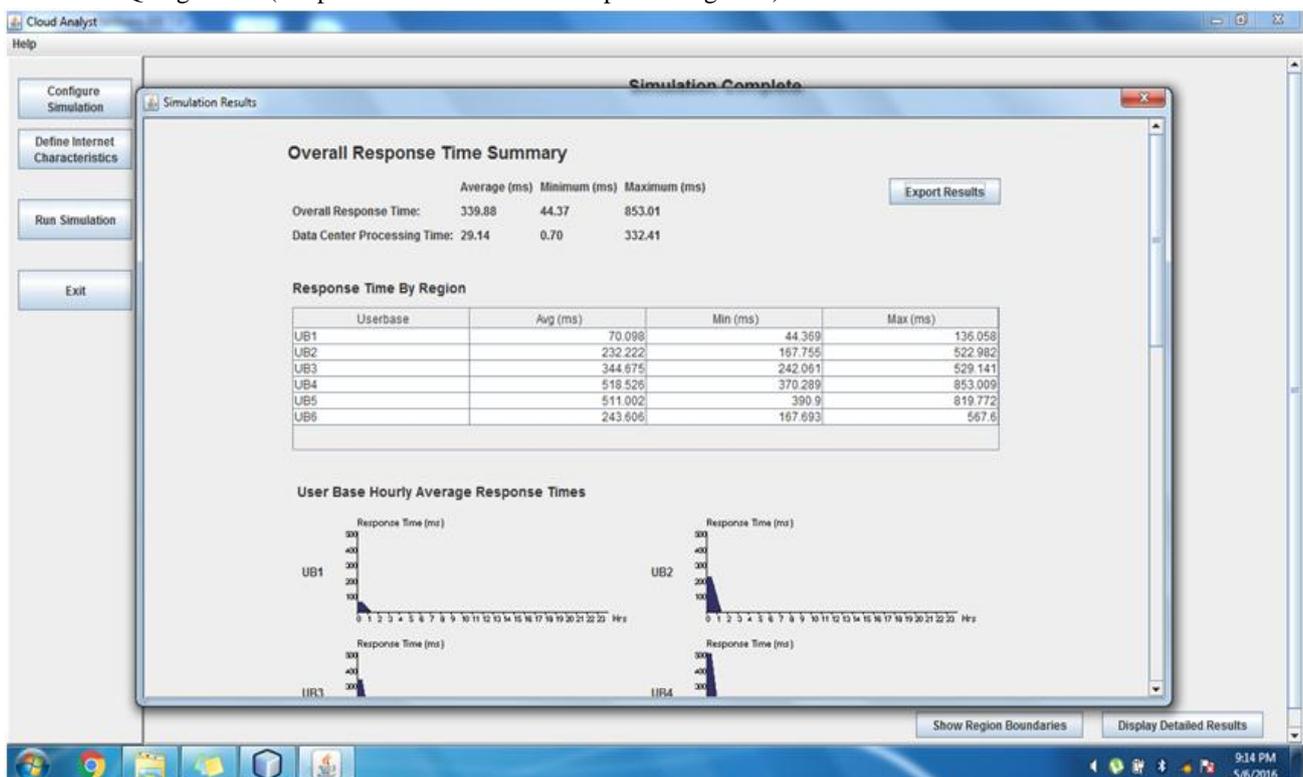


Figure4: Snapshot of Joint Ideal Queue.

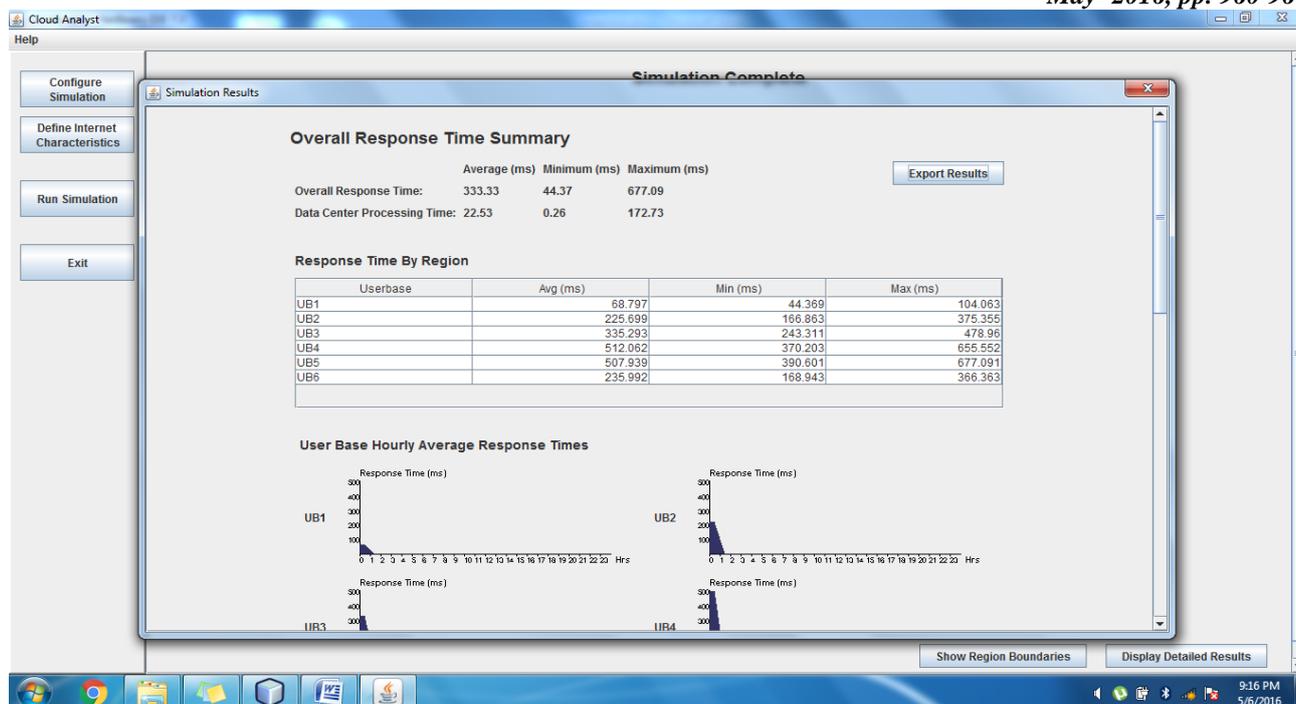


Figure5: snapshot of Modified Joint Ideal queue.

V. CONCLUSION AND FUTURE SCOPE

In this paper here we minimize the response time, datacenter response time and vm utilization time based on the JIQ and Modified JIQ based on algorithm. This implementation can be done with the help of Netbeans(IDE) and various graph can be obtain in the cloud Analyst tool. The Cloud Analyst is GUI based simulation tool built on top of CloudSim tool kit, by extending CloudSim functionality with the introduction of concepts that model Internet and Internet Application behaviors In this further work can be conducted with the help of GA optimization and also be improved with the help of HoneyBee- forging algorithm.

REFERENCES

- [1] AmandeepKaurSidhu, SupriyaKinger , “Analysis of Load BalancingTechniques in Cloud Computing ”, International Journal of Computers &Technology , Volume 4 No. 2, March-April, 2013, pp. 737-741.
- [2] Jasmin James, Dr. BhupendraVerma, “Efficient VM Load BalancingAlgorithm for a Cloud Computing Environment ”, International Journal onComputer Science and Engineering (IJCSSE) , Vol. 4 No. 09 Sep 2012 , pp.1658-1663.
- [3] Shridhar G. Domanal, G. Ram Mohana Reddy, “Load Balancing inCloud Computing Using Modified Throttled Algorithm”, IEEE,International conference. CCEM 2013.
- [4] M.Aruna, D. Bhanu, R.Punithagowri , “A Survey on Load BalancingAlgorithms in Cloud Environment ”, International Journal of ComputerApplications, Volume 82 – No 16, November 2013 , pp. 39-43.
- [5] Qi Zhang, Lu Cheng, RaoufBoutaba; Cloud computing: sate-of-artand research challenges; Published online: 20th April 2010, Copyright :The Brazillian Computer Society 2010.
- [6] Huankai Chen, Professor Frank Wang, Dr Na Helian, GbolaAkanmu,“User-Priority Guided Min-Min Scheduling Algorithm For Load Balancingin Cloud Computing”, IEEE, 2013.
- [7] Rajesh George Rajan, V.Jeyakrishna (Dec 2013), ”A Survey on Load Balancing in Cloud Computing Environment”, Vol.2, Issue.12, pp.4726-4728.
- [8] DharmeshKashyap, JaydeepViradiya(Nov 2014),”A Survey of Various Load Balancing Algorithms in Cloud Computing”, Vol.3,Issue.11,pp.115-119.
- [9] UpendraBhoi,PurviN.Ramanuj(April 2013) ”Enhanced Max-Min Task Scheduling Algorithm in Cloud Computing”, Vol.2,Issue.4,
- [10] http://en.wikipedia.org/wiki/Cloud_computing.