PROPOSED QUEUE BASED TECHNIQUE FOR LOAD BALANCING BY IMPROVING THROTTLED LOAD BALANCING ALGORITHM (PQT)

[1]Renuka Chauhan, [2]Amita Arora, [3]Mamta Mahiya
[1]Research Scholar, [2]Assistant Professor, [3]Assistant Professor
[1]CE Department,
[1]JC Bose University of Science & Technology YMCA,Faridabad, India

Abstract : Cloud Computing is becoming popular among organizations and other users day by day as it provides services easily and quickly at cheaper cost due to which there are many challenges such as providing services at minimum response time, proper resource utilization etc. has associated with it. Therefore, we have proposed a queue based approach for improving response time and resource utilization of VM on cloud computing. The simulation of the proposed technique is done on Cloud Analyst tool. The proposed algorithm has improved response time, resource utilization and virtual machines(VMs) will always be available using this approach at same cost and datacenter processing time as of existing algorithms.

IndexTerms - Cloud Computing, Load balancing, Cloud Analyst, Round Robin algorithm, ESCE algorithm, TLB algorithm, Userbases, Datacenter.

I. INTRODUCTION

A. CLOUD COMPUTING

Cloud Computing is a virtual and distributed computing model [15] which allows a large number of users to access resources over the internet. Cloud providers and cloud users are the two actors in cloud environment[18]. The resources provided are scalable. The first field that has gained its benefits is Telepsychatist[20]. The users are provided with distributed access to the resources. Cloud Computing provides services to private as well as public organizations. Some of its benefits are elasticity, pay per use and self-service provisioning [16]. Cloud Computing is providing "everything-as-a-service"[14]. Some of the services provided by cloud are listed below [5][8]:

1) Infrastructure-as-a-Service
2) Platform-as-a-Service
3) Software-as-a-Service
4) Recovery-as-a-Service
5) Function-as-a-Service

Out of all services listed above IAAS, PAAS and SAAS are the major services provided by the cloud.

B. COMPONENTS OF CLOUD COMPUTING

Cloud has three major components [10]:

1) Clients- Client component of cloud computing helps the end users to maintain their data on cloud. The end users maintain their data over cloud by interacting with the clients.

There are three types of clients-Mobile Clients, Thin Clients and Thick Clients.

2) Data Centers- The datacenter component of cloud computing can be defined as a collection of a large number of servers where application to which the end users subscribe is stored.

3) Distributed Servers- The distributed servers component of cloud computing are located/distributed all around the world and fulfills the requests from the clients whenever required.

C. MAJOR PATTERNS OF CLOUD COMPUTING [9]:

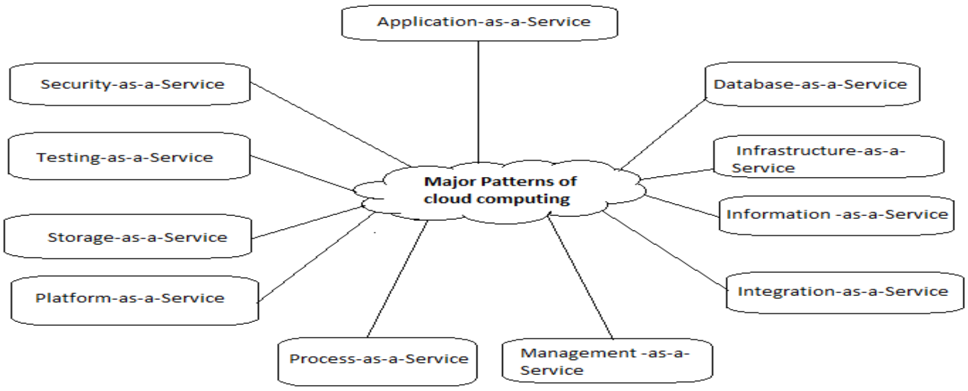

Fig. 1. Major Patterns of cloud computing

II. LITERATURE SURVEY

A. LOAD BALANCING

Load Balancing is a technique in which the workloads are distributed among resources in cloud environment to improve performance [11]. Load Balancing is the major issue associated with cloud computing. With the help of better load balancing algorithm the performance and resource utilization of the database, applications and other services can be improved[17].Balancing the load means that all the VMs are loaded equally[19].

Load Balancing is of two types:-

- Static Load Balancing: In this technique of load balancing, the work is distributed among processors before the algorithm's execution[12]. The resource utilization in this algorithm is lesser than dynamic load balancing.
- Dynamic Load Balancing: In this technique of load balancing, the work is distributed among processors during the algorithm's execution[12].

B. COMPARISON BETWEEN STATIC AND DYNAMIC LOAD BALANCING [13]

Table 1: Comparison between Static Load Balancing and Dynamic Load Balancing [13]

SNO.	Parameters	Static Load Balancing	Dynamic Load Balancing
1	Design	Easy to design	Difficult to design
2	Implementation	Easy to implement	Hard to implement
3	Response Time	Response Time is less	Response Time is more
4	Communication Delay	Less Communication Delay	More Communication Delay
5	Efficient	Efficiency is less	Efficiency is More
6	Behavior	Behavior is predictable	Behavior is unpredictable
7	Reliability	Less reliable	More reliable
8	Stability	These algorithms are more stable	These algorithms are less stable.

There are a lot of load balancing algorithms have been proposed by the researchers and some of them are discussed below:

1) Round robin algorithm(RR)

It is one of the oldest and widely used algorithms. D. Chitra Devi et al.[1] and Nguyen Xuan Phi et al.[2] had defined the round robin algorithm as RR allocates load to the virtual machines in circular motion regardless of the load on a particular Virtual machine.

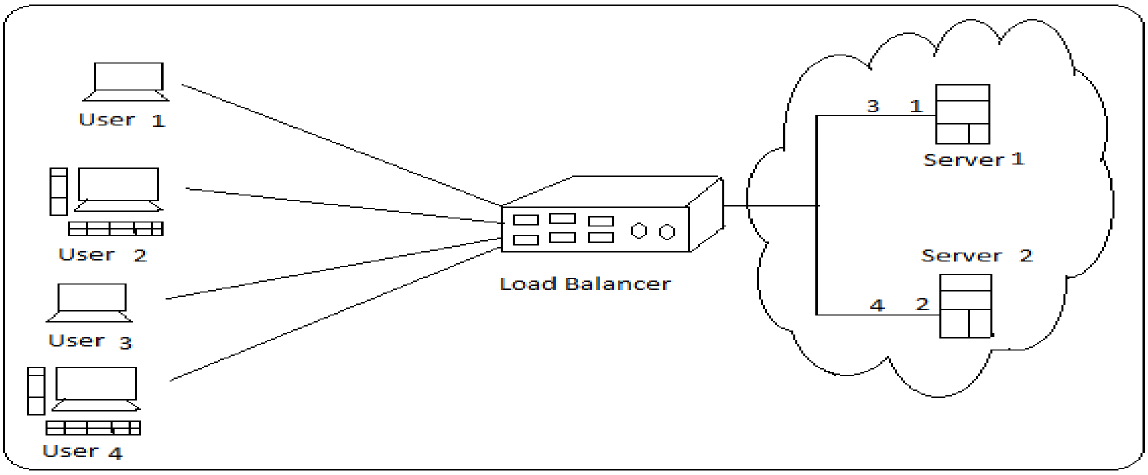

Fig.2. Round robin algorithm Flow Chart

It is based on the time quantum or time slice. In RR policy all virtual machines within the datacenter are given the very small load in circular motion regardless of their processing power, capability, length of the tasks allocated to them and priority of the tasks. In this algorithm, tasks with higher priority and large execution time suffer. It is good for datacenters with same processing power virtual machines.

Algorithm:-
INPUT: Userbases U1, U2, U3 ....Un.
Available Virtual Machines V1, V2, V3, ... Vn within datacenter.
Step1- RR algorithm consists of a ready queue which is maintained as a FIFO queue.
Step2- Each process is allocated a fixed time period known as time quantum.
Step3- The first process from the ready queue is selected and dispatched to the processor. If the process is unable to complete its task within the time quantum then it is stopped temporary with the help of the timer. The stopped process context is saved and queued back in ready queue.
Step4- The next process is selected by the scheduler and dispatched to the processor and it continues to execute until its time expires.
OUTPUT: Userbases are assigned to the virtual machines available.

Fig.3. Round robin algorithm

2) Equally Spread Current Execution Algorithm(ESCE)
Durgesh Patel et al.[3] has defined ESCE algorithm such as it attempts to distribute equal workload on each virtual machine and for that purpose it maintains an index table for all the virtual machines.

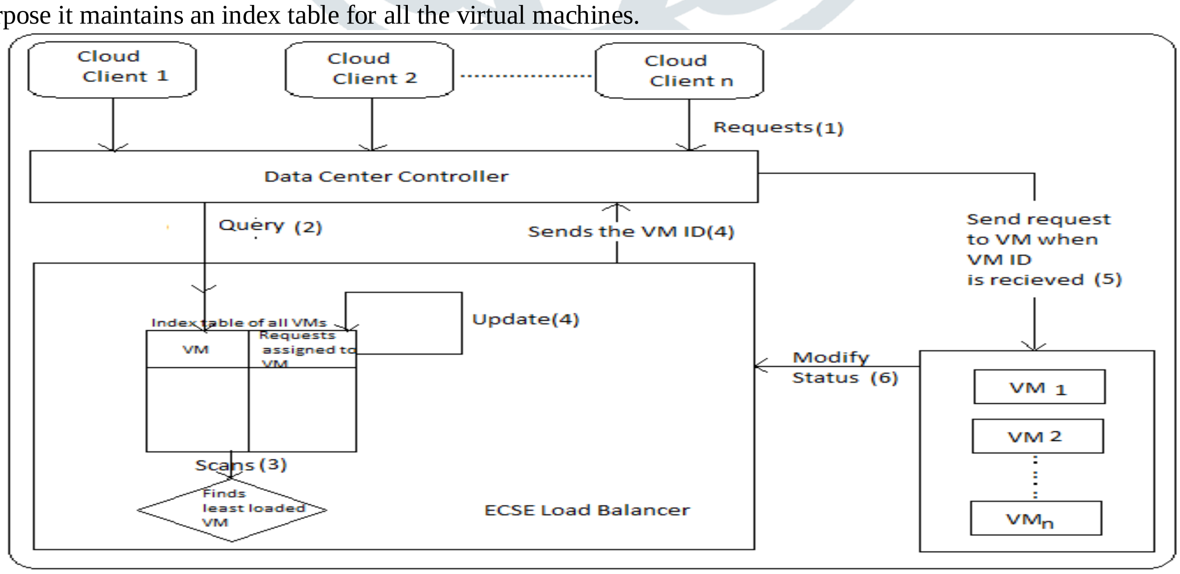

Fig.4. Equally Spread Current Execution Algorithm Flow Chart

This index table consists of virtual machine and all the requests allocated to the virtual machine. Before allocating any virtual machine the ESCE Load balancer first scans the index table to find out the virtual machine with minimum load after that the virtual machine with minimum load is allocated.

Algorithm :-
INPUT: Userbases U1, U2, U3 ....Un.
Available Virtual Machines V1, V2, V3, ... Vn within datacenter.
Step1- the cloud client sends new request to the Datacenter Controller.
Step2- ESCE load balancer is asked for next allocation by the Datacenter Controller.
Step3- The ESCE load balancer consists of an index table which maintains list of virtual machines and the request allocated to them. The ESCE checks the index table to find out the virtual machine with minimum load. If more Virtual machines are found then the virtual machine first identified is selected.
Step4- ECSE load balancer sends the ID of the virtual machine with minimum load to datacenter controller.
Step5- With the help of virtual machine ID received from the ECSE load balancer the Datacenter Controller requests the virtual machine and revises the index table by raising the count of virtual machine by one.
Step6- When the virtual machine completes its task the index table is again revised by the ESCE load balancer.
OUTPUT: Userbases are assigned to the virtual machines available.

Fig.5. Equally Spread Current Execution Algorithm

3)Throttled Algorithm

As defined by Nguyen Xuan Phi et al[2] and Tejinder Sharmaet al.[4], in this algorithm the virtual machines and their state(Available and busy) is maintained by the Throttled load balancer in an index table.

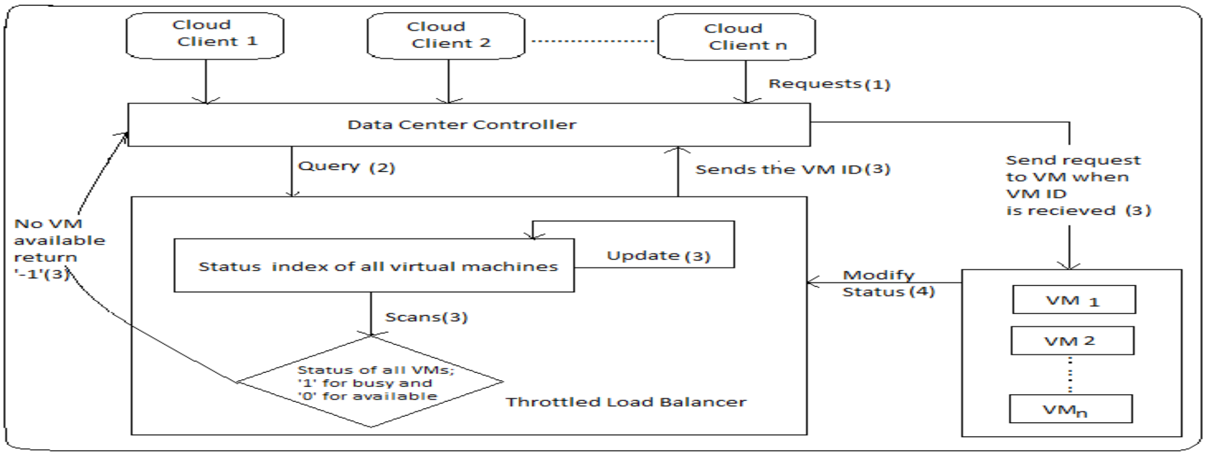

Fig.6. Throttled Algorithm Flow Chart

The virtual machine is allocated only after checking the status of the virtual machine in the index table. The working of Throttled Algorithm is discussed in the figure 6.

Algorithm:-
INPUT: Userbases U1, U2, U3 ....Un.
Available Virtual Machines V1, V2, V3, ... Vn within datacenter.
At initial the status of every virtual machine is '0' i.e. all the virtual machines are available at the start.

Step1- The cloud client sends new request to the datacenter controller.
Step2- The Throttled load balancer is asked for next allocation by the Datacenter Controller.
Step3- Now, the Throttled load balancer will scan the index table from top to get the next available virtual machine.
If virtual machine is found:
a) The virtual machine ID will be send to the datacenter controller by the Throttled load balancer.
b) The datacenter controller sends request to the virtual machine corresponding to that ID.
c) Datacenter controller informs the Throttled load balancer about a new allocation.
d) Throttled load balancer revises the index table by setting the status to '1' of the allocated virtual machine and waits for new query.
If virtual machine is not found:
a) Throttled load balancer returns '-1' to the datacenter controller.
b) Requests are arranged by the datacenter controller.
Step4- Once the task of the virtual machine is completed the datacenter controller notifies the Throttled load balancer to stop.
In case of more than one request, the datacenter controller repeats step 2 and the process continues until the index table is empty.
OUTPUT: Userbases are assigned to the virtual machines available.

Fig.7. Throttled Algorithm

4)Throttled Modified Algorithm(TMA)

This algorithm is proposed by Nguyen Xuan Phi et al. [2]. TMA is almost similar to Throttled algorithm. The only difference is that in TMA two index tables named 'Available index table' and 'Busy index table' are maintained.

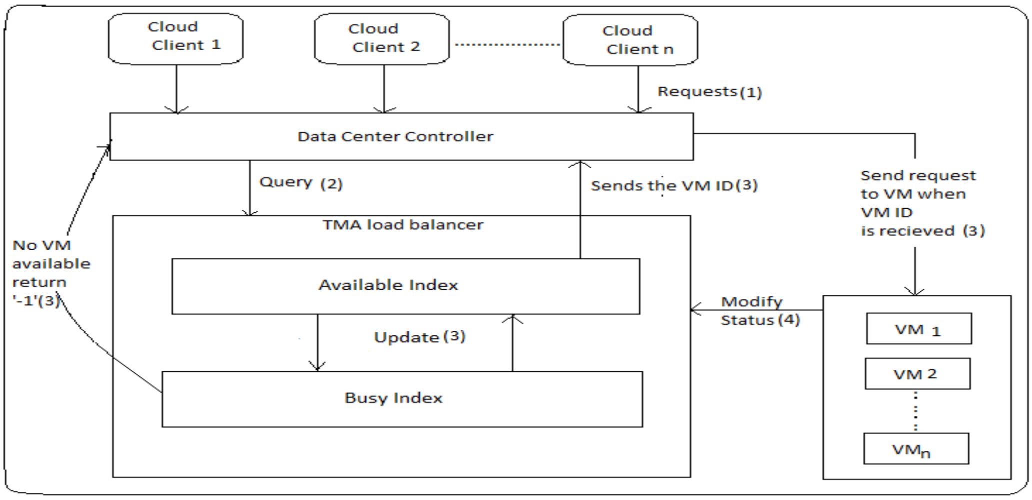

Fig.8.Throttled Modified Algorithm Flow Chart

. In this, the status if virtual machines available for allocation is '0' and the status if virtual machines unavailable for allocation is '1'.As in this algorithm separate tables are maintained; it is more flexible to identify the available virtual machines than Throttled algorithm.

Algorithm:- INPUT: Userbases U1, U2, U3 ….Un.
Available Virtual Machines V1, V2, V3, ………. Vn within datacenter.
In TMA load balancing is performed by the TMA load balancer by updating and maintaining two index tables:

- Available Index: It contains all the virtual machines which are available for allocation. The status of virtual machines is set to '0'.
- Busy index: It contains all the virtual machines which are unavailable for allocation. The status of virtual machines is set to '1'.

Initially, all the virtual machines' status is '0' i.e. all virtual machines are in 'Available index' and available for allocation and the 'Bust index' is empty.
Step1 - The cloud client sends a new request to datacenter controller.

Step2 - The TMA load balancer is asked for next allocation by the Datacenter Controller.
Step3 - TMA load balancer scans the 'Available index table' from top down and sends the ID of the virtual machine to the Datacenter Controller.

- The Datacenter Controller contacts the virtual machine identified by that ID and sends request to it.
- TMA load balancer is informed about the new allocation by the Datacenter Controller.
    - The status of this virtual machine is updated into 'Bust index table' by the TMA load balancer.

If 'Available index table' is Empty:

- '-1' value will be returned to Datacenter Controller by the TMA load balancer.
- Requests are rearranged by the Datacenter Controller.

Step4- After receiving the response from the virtual machines the Datacenter Controller informs TMA load balancer. After that the virtual machine is stopped ant 'Available index table' is updated. In case of more than one request, the datacenter controller redo step 2 and the process is continued till the 'Available index table' is empty.
OUTPUT: Userbases are assigned to the virtual machines available.

Fig.9.Throttled Modified Algorithm

5) Efficient and Enhanced Algorithm in cloud computing
This algorithm for load balancing in cloud environment by Tejinder Sharma et al. [4] is also a modified version of Throttled algorithm. The difference is that in this algorithm the researcher worked on decreasing the number of cases in which virtual machines are not available for allocation .The aim of this algorithm is to find the Expected Response Time of every virtual machine. The Efficient and Enhanced Algorithm approach helps in reducing the cost as well as the total response time. Processing time of the datacenter is also improved by this approach.

Algorithm:- INPUT: Userbases U1, U2, U3 ....Un.
Available Virtual Machines V1, V2, V3, ... Vn within datacenter.
In the beginning, the status of virtual machines in the index table is set '0' means all virtual machines are available in beginning.
Step1- The cloud client sends a new request to datacenter controller.
Step2- The TMA load balancer is asked for next allocation by the Datacenter Controller.
Step3- Datacenter Controller scans the virtual machine index table.
If virtual machine is found
- ID of virtual machine will be sent to the Datacenter Controller by the load balancer.
- Goto Step.

If virtual machine is not found:
Virtual machine list is reinitialized to '0' by using RR fashion and to identify the available virtual machine the virtual machine list is scanned in increment manner.
Step4- After receiving the expected response from virtual machine, the Datacenter Controller informs the load balancer to stop the virtual machine.
Step5- The allocation table is updated by the load balancer.
Step6- Goto Step 2.
OUTPUT: Userbases are assigned to the virtual machines available.

Fig.10. Efficient and Enhanced Algorithm

6)Efficient Throttled Load Balancing Algorithm(ETA)
ETA by Durgesh Patel et al.[3] is also a Throttled algorithm modified algorithm in which hash map table is maintained for load balancing.

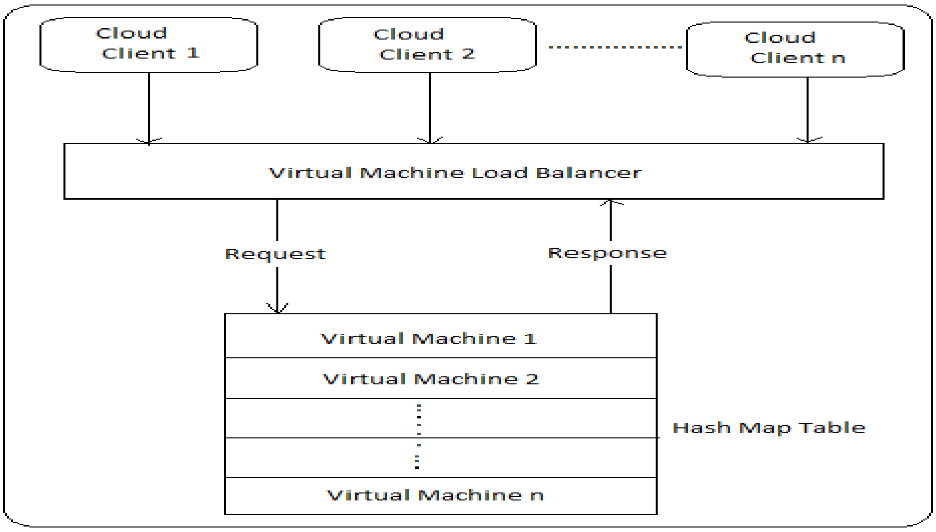

Fig.11. Efficient Throttled Load Balancing Algorithm Flow Chart

This hash map table is a table of all the virtual machines with their estimated response time and current state. The state of the virtual machine can be available or unavailable.

Algorithm:-
INPUT: Userbases U1, U2, U3 ....Un.
Available Virtual Machines V1, V2, V3, ... Vn within datacenter.
Step1- The cloud client sends a new request to datacenter controller which then queries the Efficient Throttled load balancer.
Step2- The hash map table is scanned by the efficient Throttled Algorithm.
If it found a virtual machine with minimum response time and least load then the ID of that virtual machine is sent to the Data Center Controller.

- Data Center Controller sends request to the virtual machine identified by that ID and notifies the Efficient Throttled load balancer.
- Efficient Throttled load balancer updates the virtual machine hash map table.

If virtual machine is not found: '-1' value is returned to Data Center Controller.

Step3- Data Center Controller after receiving the expected response from the virtual machine notifies the Efficient Throttled load balancer to stop the virtual machine.
Step4- Efficient Throttled load balancer updates the virtual machine hash map table.
In case of more than one request, step 1 is repeated until the Hash map table is empty.
OUTPUT: Userbases are assigned to the virtual machines available.

Fig.12. Efficient Throttled Load Balancing Algorithm

III. PROPOSED QUEUE BASED TECHNIQUE FOR LOAD BALANCING BY IMPROVING THROTTLED LOAD BALANCING ALGORITHM (PQT)

Step 1:- Initialize the VMs status table, VMCounter Table and Dispatcher Queue for each VM.
Step 2:- Put all available VMs to the VMCounter and Dispatcher Idle Queue using "Enqueue policy".
Step 3:- Load Balancer initiates-
I. Status of VM to 'Available'.
II. Initially sets counter of every VM to '0'.
III. Distributes VM to dispatchers.
Step 4:- When there is request for VM, the load balancer call "Pick idle VM" method to select idle VM using dequeue policy. Dequeue Policy Works as follows:
I. Select dispatcher with maximum idle queue length.
II. Pick first VM from selected dispatchers' idle queue and return selected VM ID to controller.

Step 5:- Change selected VMs state to BUSY and increment the VN counter by one in VMCounter table.
Step 6:- When selected VM is free then update the state of VM to AVAILABLE and put that VM to dispatcher by calling "add idle VM" method using selection-policy in Enqueue technique. Enqueue Policy is as follows:

a) If selection policy is "RANDOM" then
I. Select dispatcher randomly.
II. Put idle VM to end of idle queue of selected dispatcher.

b) If selection policy is "SQ" then
I. Select dispatcher with minimum queue length.
II. Append VM to end of idle queue of selected dispatcher.
END

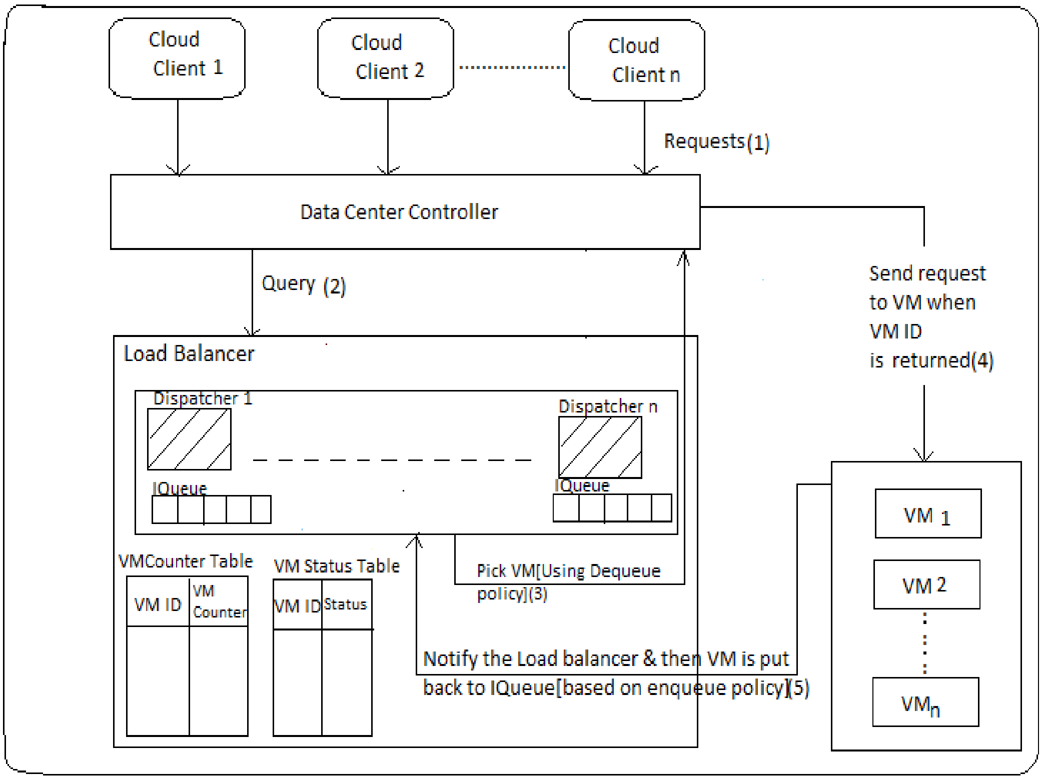

Fig.13. Proposed Queue Load Balancing Technique Flow Chart

IV. SIMULATION SETUP AND PERFORMANCE ANALYSIS

To compare the performance of the proposed Queue Load Balancing Technique with the existing Round robin, TLB and Equally spread current execution techniques we use Cloud Analyst tool. Cloud Analyst is a GUI based simulation tool and is built on a toolkit called Cloudsim[3][6][7]. It allows simulation and other experimentations. With Cloud Analyst the location of the users and data centers can be configured easily and various parameters such as number of userbases, Datacenters, Service Broker Policy, Simulation Duration, Application deployment configuration, User grouping factor in userbases, Request grouping factor in datacenters, Executable instructions length per request and load balancing policy across VMs in a single datacenter can also be configured easily and quickly on this tool. It allows to save the configurations. Figure below shows the GUI of Cloud Analyst.

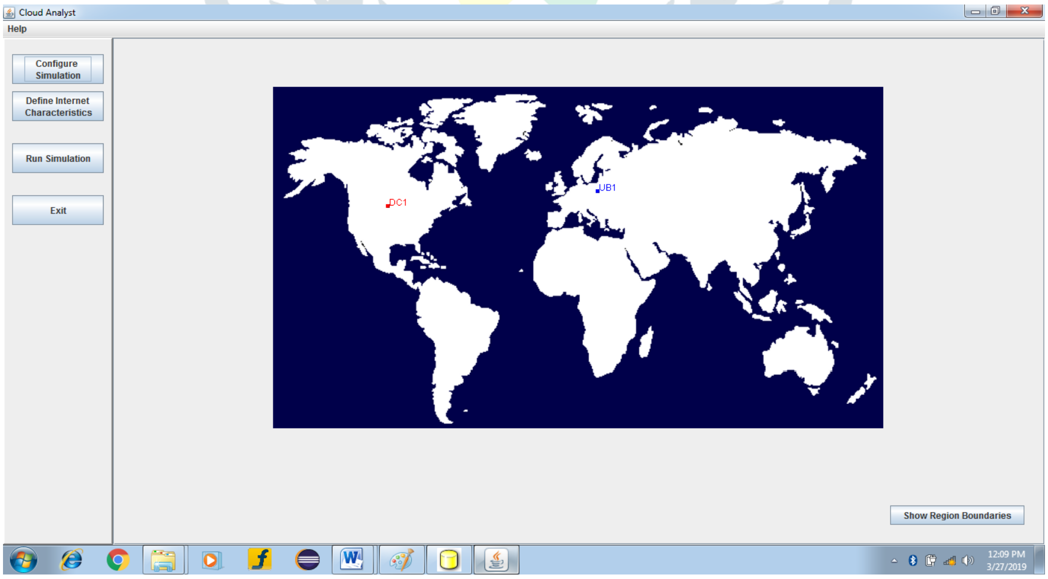

Fig14. Output screen of Cloud Analyst

To set the whole simulation process we use three main features –

i) Configure Simulation
ii) Define Internet Characteristics
iii) Run Simulation

In this paper four scenarios are carried out to compare the performance of Proposed Queue Load Balancing Technique with the existing Round robin, TLB and Equally spread current execution techniques based on response time, datacenter processing time, cost and resource utilization. As in this paper we are using random policy, so for each load balancing technique an average of five times reading is calculated in order to perform comparison between them in every scenario. Following tables shows the parameters which are fixed in all the scenarios.

Table 2: PARAMETERS CONFIGURATION FOR DATACENTERS

	PARAMETER	VALUES
1.	VM Memory(Mb)	204800
2.	VM Storage(Mb)	100000000
3.	VM Available BW	1000000
4.	Number of processors in VM	4
5.	Processor speed	10000
6.	VM policy	Time_Shared
7.	Datacenter Architecture	X86
8.	Datacenter OS	Linux
9.	Datacenter VMM	Xen
10.	Datacenter Cost per VM $/H	0.1
11.	Datacenter memory cost $/H	0.05
12.	Datacenter Storage cost $/S	0.1
13.	Datacenter data transfer cost $/Gb	0.1
14.	Datacenter physical H/W units	2
15.	VMS in every datacenter	5

Table 3: ADVANCED SETTINGS

1.	User grouping factor in Userbases	10
2.	Request grouping factor in Datacenters	10
3.	Executable instruction length per request(bytes)	100

Scenario 1:
In this scenario there are five userbases UB1, UB2, UB3, UB4 and UB5 at region 0,1,2,3 and 4 respectively and one datacenter DC1 at region 5. The Userbases and datacenter are at different locations. The numbers of dispatchers used in PQT are two. The datacenter DC1 has 5 VMs. The reading is calculated five times and the average of five readings is used to compare all the techniques. Other parameters are same in every load balancing technique as specified in table 1 and 2. The following Table 4 shows comparison between Round Robin , ESCE, TLB and PQT in terms of response time for this scenario.

Table 4: Comparison between Round Robin , ESCE, TLB and PQT in terms of Response Time:

	Round Robin	ESCE	TLB	PQT
1.	601.12	601.13	601.16	601.08
2.	601.09	601.18	601.11	601.10
3.	601.11	601.13	601.11	601.12
4.	601.14	601.17	601.17	601.24
5.	601.14	601.24	601.15	601.05
Average	601..120	601.170	601.140	601.118

The table clearly indicates that Proposed Queue Technique has minimum average response time than Round Robin, ESCE and TLB algorithm.The following Table 5 shows comparison between Round Robin , ESCE, TLB and PQT in terms of cost for this scenario.

Table 5: Comparison between Round Robin, ESCE, TLB and PQT in terms of Cost:

	Round Robin	ESCE	TLB	PQT
Cost($)	0.82	0.82	0.82	0.82

The table shows that Proposed Queue Technique charges the same cost as charged by Round Robin, ESCE and TLB algorithm with the advantage that it processes the query in less time as compare to other algorithms.The following Table 6 shows comparison between Round Robin , ESCE, TLB and PQT in terms of datacenter processing time for this scenario.

Table 6: Comparison between Round Robin , ESCE, TLB and PQT in terms of Datacenter Processing Time:

	Round Robin	ESCE	TLB	PQT
Datacenter Processing Time	0.10	0.10	0.10	0.10

The table 6 shows that the PQT performs better than Round Robin , ESCE and TLB in terms of response time, resource utilization and "no VM available" condition will never occur in PQT with same datacenter processing time. The following Table 7 shows comparison between Round Robin , ESCE, TLB and PQT in terms of resource utilization for this scenario.

Table 7: Comparison between Round Robin , ESCE, TLB and PQT in terms of Resource Utilization:

	Round Robin	ESCE	TLB	PQT
VM0	1274	5919	5919	1275
VM1	1274	397	397	1276
VM2	1274	51	51	1273
VM 3	1274	3	3	1273
VM4	1274	3		1273

The table shows the comparison among between Round Robin, ESCE, TLB and PQT in terms of resource utilization. The table 7 clearly indicates that in proposed Queue Technique and Round Robin algorithm every virtual machine is utilized evenly whereas in ESCE and TLB more work is given to the VM which is located near to the userbase which leads to underutilization of other resources.

Scenario 2:

In this scenario there are ten userbases UB1, UB2, UB3, UB4, UB5, UB6, UB7, UB8, UB9 and UB10 at region 0,1,2,3,3,4,2,0,1 and 2 respectively and one datacenter DC1 at region 5. The Userbases and datacenter are at different locations. The numbers of dispatchers used in PQT are two. The datacenter DC1 has 5 VMS. The reading is calculated five times and the average of five readings is used to compare all the techniques. Other parameters are same in every load balancing technique as specified in table 1 and 2. The following Table 8 shows comparison between Round Robin , ESCE, TLB and PQT in terms of response time for this scenario.

Table 8: Comparison between Round Robin, ESCE, TLB and PQT in terms of Response Time:

	Round Robin	ESCE	TLB	PQT
1.	540.34	540.38	540.29	540.34
2.	540.45	540.31	540.32	540.35
3.	540.40	540.38	540.33	540.29
4.	540.38	540.33	540.34	540.33
5.	540.40	540.30	540.39	540.32
Average	540.394	540.340	540.334	540.326

The table clearly indicates that Proposed Queue Technique has minimum average response time than Round Robin, ESCE and TLB algorithm. The following Table 9 shows comparison between Round Robin , ESCE, TLB and PQT in terms of cost for this scenario.

Table 9: Comparison between Round Robin , ESCE, TLB and PQT in terms of Cost:

	Round Robin	ESCE	TLB	PQT
Cost($)	1.14	1.14	1.14	1.14

The table shows that Proposed Queue Technique charges the same cost as charged by Round Robin, ESCE and TLB algorithm with the advantage that it processes the query in less time as compare to other algorithms. The following Table 10 shows comparison between Round Robin , ESCE, TLB and PQT in terms of datacenter processing time for this scenario.

Table 10: Comparison between Round Robin , ESCE, TLB and PQT in terms of Datacenter Processing Time:

	Round Robin	ESCE	TLB	PQT
Datacenter Processing Time	0.10	0.10	0.10	0.10

The table 10 shows that the PQT performs better than Round Robin , ESCE and TLB in terms of response time, resource utilization and "no VM available" condition will never occur in PQT with same datacenter processing time.
The following Table 11 shows comparison between Round Robin , ESCE, TLB and PQT in terms of resource utilization for this scenario.

Table 11: Comparison between Round Robin , ESCE, TLB and PQT in terms of Resource Utilization:

	Round Robin	ESCE	TLB	PQT
VM0	2550	11808	11799	2549
VM1	2549	835	842	2549
VM2	2549	91	95	2549
VM 3	2549	10	7	2549
VM4	2549	2	3	2550

The table shows the comparison among between Round Robin, ESCE, TLB and PQT in terms of resource utilization. The table 11 clearly indicates that in proposed Queue Technique and Round Robin algorithm every virtual machine is utilized evenly whereas in ESCE and TLB more work is given to the VM which is located near to the userbase which leads to underutilization of other resources.

Scenario 3:

In this scenario there are five userbases UB1, UB2, UB3, UB4 and UB5 at region 1,2,3,4 and 5 respectively and two datacenters DC1 and DC2 at region 5. The Userbases and datacenter are at different locations. The numbers of dispatchers used in PQT are two. The datacenters DC1 and DC2 have 2 physical hardware units and 5 VMs. The reading is calculated five times and the average of five readings is used to compare all the techniques. Other parameters are same in every load balancing technique as specified in table 1 and 2. The below Table 12 shows comparison between Round Robin , ESCE, TLB and PQT in terms of response time for this scenario.

Table 12: Comparison between Round Robin , ESCE, TLB and PQT in terms of Response Time:

	Round Robin	ESCE	TLB	PQT
1.	338.19	338.12	338.20	338.16
2.	338.16	338.16	338.13	338.15
3.	338.11	338.17	338.18	338.15
4.	338.19	338.18	338.17	338.17
5.	338.13	338.14	338.13	338.12
Average	338.156	338.154	338.162	338.150

The table clearly indicates that Proposed Queue Technique has minimum average response time than Round Robin, ESCE and TLB algorithm. The following Table 13 shows comparison between Round Robin , ESCE, TLB and PQT in terms of cost for this scenario.

Table 13: Comparison between Round Robin , ESCE, TLB and PQT in terms of Cost:

	Round Robin	ESCE	TLB	PQT
Cost($)	1.32	1.32	1.32	1.32

The table shows that Proposed Queue Technique charges the same cost as charged by Round Robin, ESCE and TLB algorithm with the advantage that it processes the query in less time as compare to other algorithms. The following Table 14 shows comparison between Round Robin , ESCE, TLB and PQT in terms of co datacenter processing time st for this scenario.

Table 14: Comparison between Round Robin, ESCE, TLB and PQT in terms of Datacenter Processing Time:

	Round Robin	ESCE	TLB	PQT
Datacenter Processing Time	0.25	0.25	0.25	0.25

The table 6 shows that the PQT performs better than Round Robin, ESCE and TLB in terms of response time, resource utilization and “no VM available” condition will never occur in PQT with same datacenter processing time. The following Table 15 shows comparison between Round Robin , ESCE, TLB and PQT in terms of resource utilization for this scenario.

Table 15: Comparison between Round Robin , ESCE, TLB and PQT in terms of Resource Utilization:

	Round Robin		ESCE		TLB		PQT	
	DC1	DC2	DC1	DC2	DC1	DC2	DC1	DC2
VM0	630	646	3146	2980	3177	2969	641	635
VM1	630	646	115	106	108	111	642	635
VM2	630	646	8	8	5	6	641	635
VM 3	629	645	3	3	0	0	639	636
VM4	629	645	2	2	0	0	639	633

The table shows the comparison among between Round Robin, ESCE, TLB and PQT in terms of resource utilization. The table 15 clearly indicates that in proposed Queue Technique and Round Robin algorithm every virtual machine is utilized evenly whereas in ESCE and TLB more work is given to the VM which is located near to the userbase which leads to underutilization of other resources.

- In this technique " VM not available condition" can never occur as we have created an additional "VMCounter table" which in backend keeps record of number of times a VM has been used and every time whenever a VM is required the load balancer checks the table and allocate the VM with least counter among all. Therefore, in the proposed queue technique VM will be created every time.

V. EXPERIMENTAL RESULTS

A. Scenario 1:

The graph in the following figure 15 clearly shows that for scenario 1 where there are five userbases and one datacenter, the Proposed Queue based Technique performs better than Round Robin, Equally spread current execution and Throttled load balancing algorithm in terms of response time with same datacenter processing time.

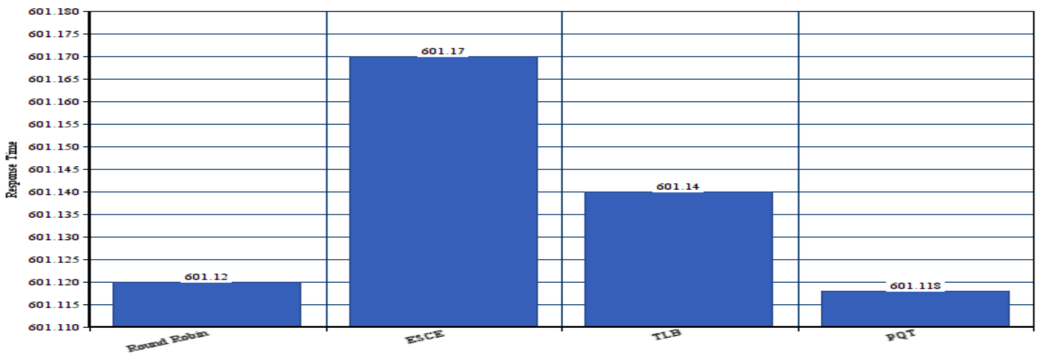

Fig.15. Analytical comparison of average response time for scenario 1

The graph in the following figure 16 clearly shows that the Proposed Queue based Technique performs better that Equally spread current execution and Throttled load balancing algorithms in terms of resource utilization with same datacenter processing time. PQTs' resource utilization is almost similar in Round robin but with better response time.

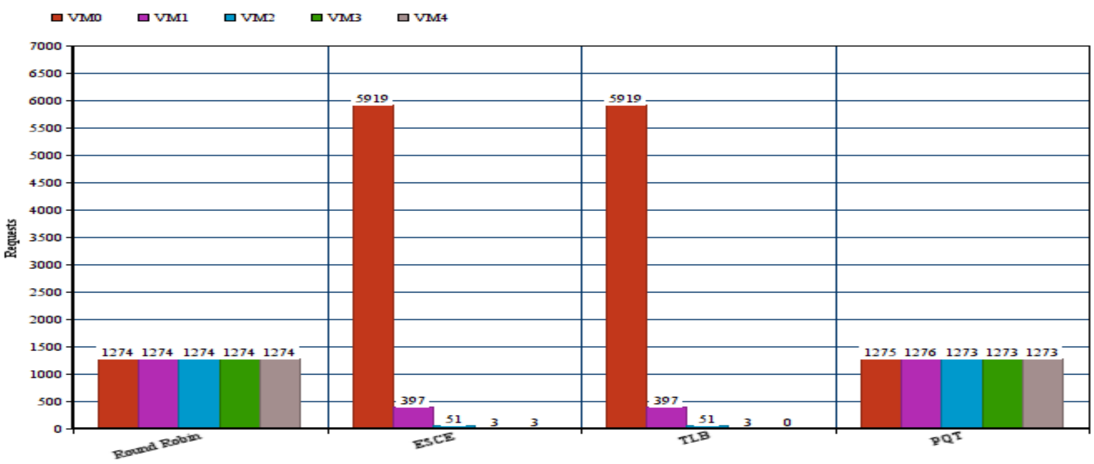

Fig.16. Analytical comparison of Resource Utilization for scenario 1

B. Scenario 2:

The graph in the following figure 17 clearly shows that for scenario where there are ten userbases and two datacenter, the Proposed Queue based Technique performs better than Round Robin, Equally spread current execution and Throttled load balancing algorithm in terms of response time with same datacenter processing time.

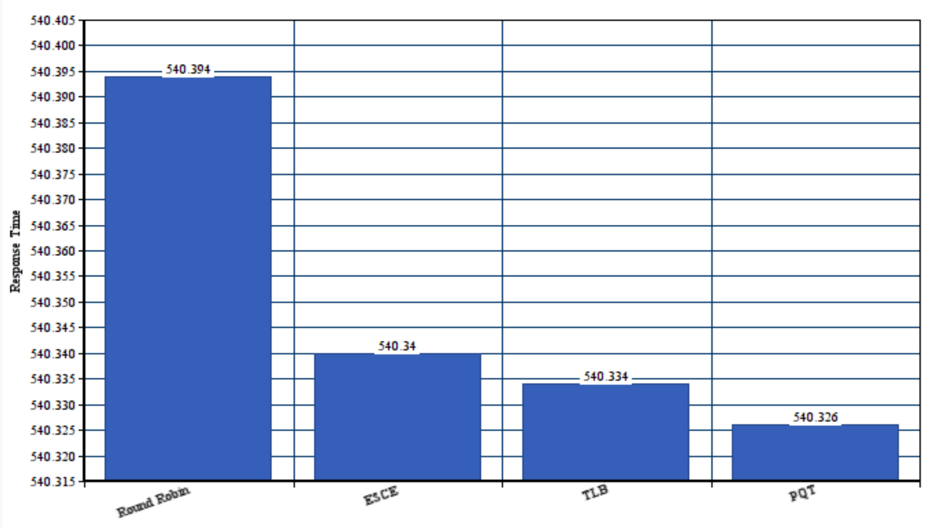

Fig.17. Analytical comparison of average response time for scenario 2

The graph in the following figure 18 clearly shows that the Proposed Queue based Technique performs better that Equally spread current execution and Throttled load balancing algorithms in terms of resource utilization with same datacenter processing time. PQTs' resource utilization is almost similar in Round robin but with better response time.

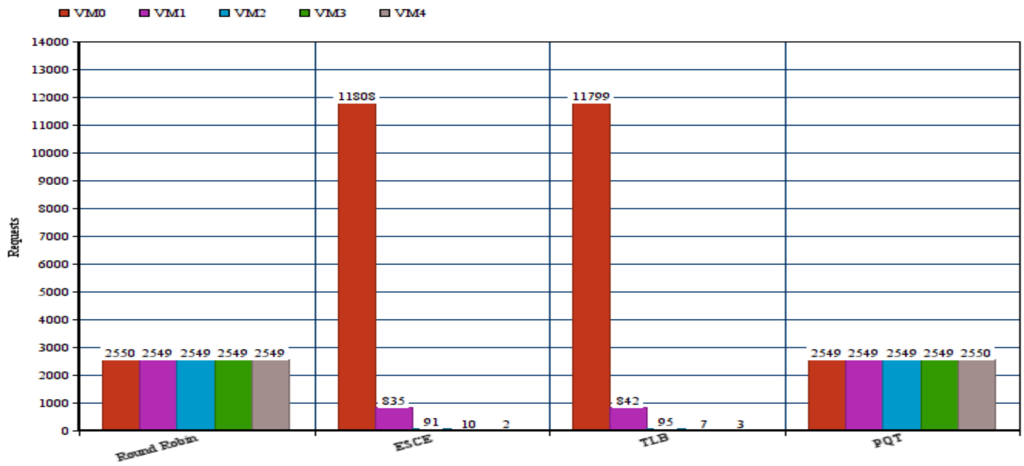

Fig.18. Analytical comparison of Resource Utilization for scenario 2

C. Scenario 3:

The graph in the following figure 19 clearly shows that for scenario 3 where there are five userbases and two datacenter, the Proposed Queue based Technique performs better than Round Robin, Equally spread current execution and Throttled load balancing algorithm in terms of response time with same datacenter processing time.

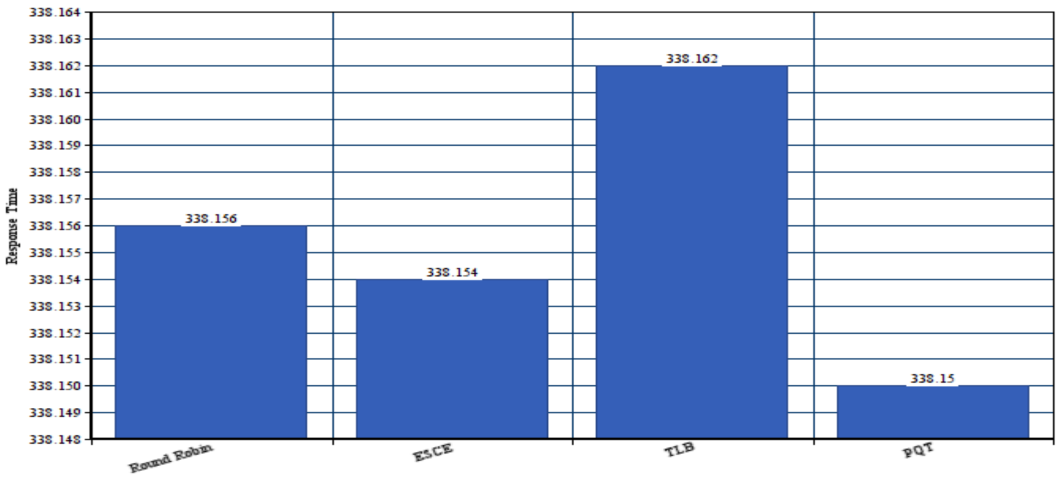

Fig.19. Analytical comparison of average response time for scenario 3

The graph in the following figure 20 clearly shows that the Proposed Queue based Technique performs better that Equally spread current execution and Throttled load balancing algorithms in terms of resource utilization with same datacenter processing time. PQTs' resource utilization is almost similar in Round robin but with better response time.

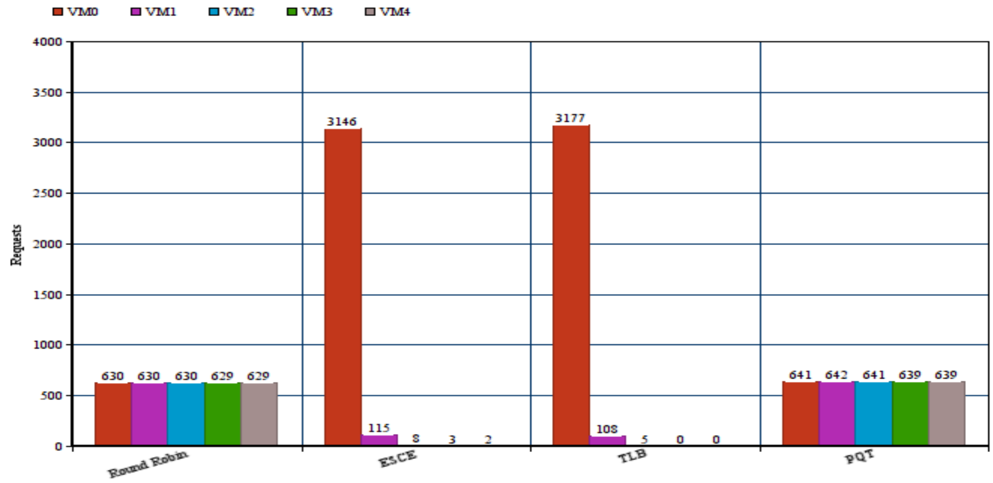

Fig.20. Analytical comparison of Resource Utilization for DC1 in scenario 3

The graph in the following figure 21 clearly shows that the Proposed Queue based Technique performs better that Equally spread current execution and Throttled load balancing algorithms in terms of resource utilization with same datacenter processing time. PQTs' resource utilization is almost similar in Round robin but with better response time.

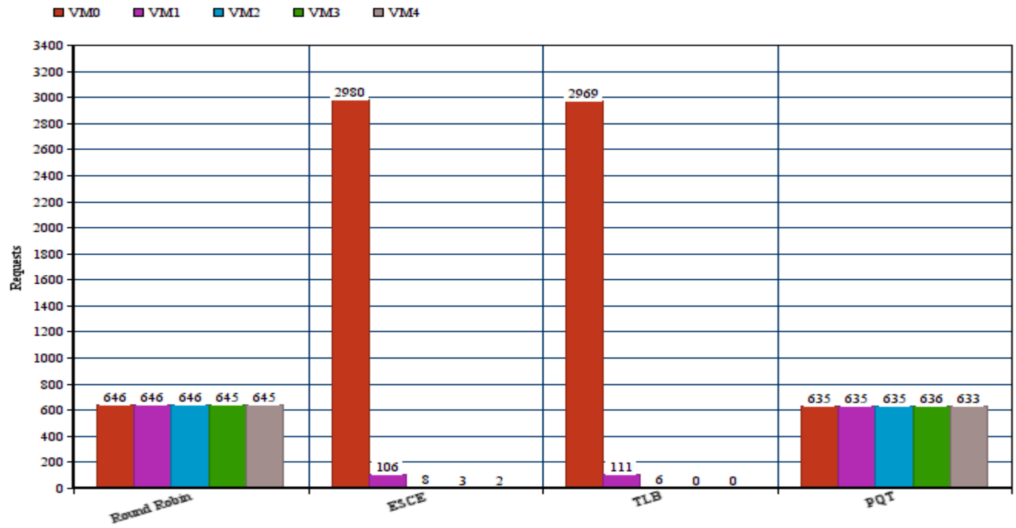

Fig.21. Analytical comparison of Resource Utilization for DC2 in scenario 3

VI. CONCLUSION AND FUTURE WORK

Proposed Queue Based Technique for load balancing by improving Throttled load balancing algorithm (PQT) is proposed in this paper. The proposed algorithm is implemented in Cloud Analyst simulation tool and the results of different scenarios are depicted in graph from which we can easily conclude that the response time of proposed algorithm is better than Round robin, ESCE and TLB algorithms and resource utilization is better than ESCE and TLB algorithms and similar to round robin with same cost and datacenter processing time. Also "VM not available" condition will never occur in the proposed algorithm (i.e. VMs will always be available) which was the limitation of some algorithms. The future work involves decreasing the cost and datacenter processing time.

References

[1] D. Chitra Devi and V.Rhymend Uthariaraj " Load Balancing in Cloud Computing Environment using Improved Weighted Round Robin Algorithm for Non-preemptive Dependent Task" , The Science world journal volume 2016.

[2] Nguyen Xuan Phi, Cao Trung Tin, Luu Nguyen Ky Thu and Tran Cong Hung "PROPOSED LOAD BALANCING ALGORITHM TO REDUCE RESPONSE TIME AND PROCESSING TIME ON CLOUD COMPUTING", International Journal of Computer Networks & Communications (IJCNC) Vol.10, No.3, May 2018, Posts and Telecommunications Institute of Technology, Ho Chi Minh, Vietnam.

[3] Durgesh Patel1 , Mr. Anand S Rajawat2 "Efficient Throttled Load Balancing Algorithm in Cloud Environment", International Journal of Modern Trends in Engineering and Research, 1 P. G. Scholar, Department of CSE, SVITS Indore, M.P., India ,2Assistant Professor, Department of CSE, SVITS Indore, M.P., India

[4] Tejinder Sharma and Vijay Kumar Banga "Efficient and Enhanced Algorithm in Cloud Computing", IJSCE Vol 3,Issue-1, March 2013.

[5] https://medium.com/@nnilesh7756/what-are-cloud-computing-services-iaas-caas-paas-faas-saas-ac0f6022d36e.

[6] Soumya Ranjan Jena Asst. Professor M.I.E.T Dept of CSE Bhubaneswar, Zulfikhar Ahmad Asst. Professor M.I.E.T Dept of CSE Bhubaneswar, "Response Time Minimization of Different Load Balancing Algorithms in Cloud Computing Environment", International Journal of Computer Applications (0975 – 8887) Volume 69– No.17, May 2013.

[7] cloudbus.org/students/MEDC_Project_Report_Bhathiya_318282.pdf.

[8] https://en.wikipedia.org/wiki/Recovery_as_a_service.

[9] https://www.quora.com/What-are-the-components-of-cloud-computing.

[10] http://techaddas.blogspot.com/2013/05/basic-components-of-cloud-computing.html.

[11] https://en.wikipedia.org/wiki/Cloud_load_balancing.

[12] Saleh A. Khawatreh Dept. of Computer Engineering, Faculty of Engineering, Al-Ahliyya Amman University Amman-Jordan,"An Efficient Algorithm for Load Balancing in Multiprocessor Systems", (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 9, No. 3, 2018.

[13] Mr. Ramesh Prajapati, Mr. Dushyantsinh Rathod and Dr. Samrat Khanna, "COMPARISON OF STATIC AND DYNAMIC LOAD BALANCING IN GRID COMPUTING", International Journal For Technological Research In Engineering Volume 2, Issue 7, March-2015 ISSN (Online): 2347 - 4718 www.ijtre.com Copyright 2015.All rights reserved. 1337.

[14] Manjeet Singh, School of Computing Science and Engineering, Galgotias University, G. Noida, U.P., India, "STUDY ON CLOUD COMPUTING AND CLOUD DATABASE", International Conference on Computing, Communication and Automation (ICCCA2015).

[15] Sapna andnM Afshar Alam, " Comparative Study of Traditional Database and Cloud Computing Database ",Volume 8, No. 2, March 2017 (Special Issue) International Journal of Advanced Research in Computer Science.

[16] Maha A. Sayal1, Ehsan Ali Al-Zubaidi2 ,1Computer Science Department, College of Computer Science & Mathematics, University of Thi-Qar Thi-Qar, IRAQ, 2Computer and Internet Department, Faculty of Physical Planning, University of Kufa, Najaf,IRAQ, "Building Web Application Using Cloud Computing ",Sayal et al., International Journal of Software and Web Sciences, 18(1), September-November, 2016, pp. 01-05.

[17] Shalini Joshi,Computer Science & Engineering, Mody University of Science and Technology, Lakshmangarh,India and Uma Kumari, Computer Science & Engineering, Mody University of Science and Technology, Lakshmangarh, India, "Load balancing in cloud computing: Challenges & issues", 2016 2nd International Conference on Contemporary Computing and Informatics (IC3I).

[18] Syed Hamid Hussain Madni (2016), "Recent advancements in resource allocation techniques for cloud computing environment: a systematic review", Springer Science+Business Media New York 2016, DOI 10.1007/s10586-016-0684-4.

[19] Mehak Choudhary, Dimple Chandra and Dr. Deepti Gupta,"Load Balancing Algorithm using JIQ methodology for Virtual Machines", International Conference on Computing, Communication and Automation (ICCCA2017).

[20] Ms. PrincyMatlani and Dr. Narendra D Londhe, Member, IEEE," A CLOUD COMPUTING BASED TELEMEDICINE SERVICE", 2013 IEEE Point-of-Care Healthcare Technologies (PHT) Bangalore, India, 16 - 18 January, 2013.