International Journal of
Advances in Scientific Research and Engineering (ijasre)

E-ISSN : 2454-8006

DOI: 10.31695/IJASRE.2019.33430

Volume 5, Issue 7
July - 2019

Performance Analysis of a Hybrid Approach to Enhance Load Balancing in a Heterogeneous Cloud Environment

Abdullahi Nafisatu Aliyu[1] and Professor Boukari Souley[2]

Department Of Mathematical Science

Abubakar Tafawa Balewa University

Bauchi,Bauchi

Nigeria

ABSTRACT

Cloud computing is rapidly becoming more advanced. It has become more attractive due to its cloud potentials of being easy to use and anywhere accessibility in comparison to other technologies. Load balancing is a very important component of efficient operations in the cloud computing environment. Several algorithms have been designed but yet unable to rectify the holes found in the performance area like minimizing response time, processing time and cost. In this work, it is proposed a hybrid approach that is devoted to enhancing performance for the cloud system user putting the concepts of enhanced Throttled and Equally spread current execution algorithms(ESCE) into use. The proposed hybrid algorithm was implemented in cloud analyst environment and the results obtained were analyzed. The performance of the proposed hybrid algorithm was evaluated with the existing algorithms ESCE and the improved throttled using the response time, data processing time, cost, physical memory and storage capacity. The results of the evaluation analysis show that with network delay, physical memory and the storage capacity, the Performance differences in various scenario vary, where scenario 1 is 10.5 (ms) average response time and processing time, $ 2.10 total cost. Scenario 2 ,5.6 (ms) response time and 8.5(ms)processing time $2.03 cost but the ESCE recoded best minimum response time of 40.30(ms). Scenario3,8.3 (ms) response time, 3.5 (ms)processing time,$2.00 cost. scenario 4 of 0.31(ms) response time, the processing time of 2.41(ms) and cost of $2.14 but the improved throttled is best in terms of minimum processing time with 2.00(ms).scenario 5,2.3 (ms) response time ,processing time 1.1 (ms)and cost of $2.00. Considering the physical memory, the fixed physical memory provided a better average response time of 0.50(ms) than the varied physical memory, and the varied physical memory provides a better maximum processing time of 5.03(ms). In terms of storage capacity, high storage capacity provides a better response time of 320.02 (ms). This work recommends the adaptation of hybridized improved Throttled and ESCE algorithms by cloud service providers so as to get better performances.

Key Words: *Cloud Computing, Load Balancing, Performance, Cloud Analyst, Algorithm.*

1 INTRODUCTION

1.1 Cloud Computing

Cloud enables us to access information from anywhere at any giving time. Cloud computing is a type of computing concepts that involve a very large number of computers that are connected through a network such as internet. [10] Heterogeneous environment means having different hardware characteristics including, memory, storage ,CPU(central processing unit), bandwidth and other hardware and software components. (Sharma and Banga 2013).Cloud Computing is an upcoming computing technology. It is there to share data, make several calculations, and service transparently over a scalable network of nodes. [8] A computing system maintains a number of interconnected virtual machines to process various tasks from different users called cloud. Cloud computing consists of three major components which are Datacenters, servers and VM (virtual machines). A cloud is made up of multiple datacenters, each datacenter contains group of servers and each server is extended by different number of virtual machines [4].

1.2 Load Balancing

Load balancing is a very important concept in network. The load balancer takes different requests from the client and distribute each of them across multiple computers or network devices depending on how busy the computer or network device is [7].The major goals of load balancing algorithms are:

1. To achieve a whole improvement in system performance at a reasonable cost [15].
2. To maintain a backup plan in a situation where the system fails even partially [17].
3. To enable future modification in the system: the distributed system can change such as applying new topology and scale up. So a load balancing algorithm must have to be scalable and flexible to handle such changes.

1.3 Types of Load Balancing Algorithms

1.3.1 Static Load Balancing

In this type of load balancing algorithm the decision of shifting the load does not base solely on the current state of the system. It needs knowledge on the applications and resources of the system. The time at which job arrives determines the performance of the virtual machines . The master processor allocate the workload to other slave processors based on their performance. The allotted work is thus performed by the slave processors and the result is returned to the master processor.

1.3.2 Dynamic Load Balancing

This type of load balancing algorithms make use of the current state of the system to make any kind of decision for load balancing, the current state of the system also takes control of the shifting of the load . It allows for processes to move from an over utilized machine to an under-utilized machine dynamically for faster execution. this approach important because its decision for balancing the load is based on the current state of the system which helps in improving the overall performance of the system by migrating the load dynamically. [13]

1.4 Existing Load Balancing Algorithms

1.4.1 Equally Spread Current Execution Algorithm (ESCE).

[3]. The load balancer tries to preserve equal load to all the virtual machines connected with the data centre. In Equally spread current execution algorithm, the processes are handled with load priorities. It distributes the load to virtual machine by checking the load at current time and transfer of the load to that virtual machine which is lightly loaded and handles that request easily and result in less time taken , and give maximum possible throughput It is spread spectrum technique in which the load balancer spreads the load into multiple virtual machines.

1.4.2 Throttled Load Balancing Algorithm (TLB)

[19] In TLB algorithm, an index table is maintained by load balancer which contains virtual machines as well as their states (Available or Busy). On receiving a request from client data centre firstly tries to find a suitable virtual machine (VM) to perform the requested task. The data centre broker queries the load balancer for allocation of the VM. The load balancer scans the index table from top until the first available VM is found or the index table is scanned fully. If the status of any VM is Available, then VM id is send to the data centre. The data centre then allocates the request to the VM identified using the throttled algorithm. Also, the data centre updates the index table and set the state of VM to busy. But during processing the request of client, if no VM is found, the load balancer returns -1 to the data centre.

1.4.3 Improved Throttled Algorithm

The improved throttled Algorithm by [6] is devoted to performance enhancement for the user of the existing cloud system by improving the basic throttled mapping approach between task and resources. the improved throttled algorithm was achieved through the use of priority. by setting priority to each VM. The Priority is calculated based on the capacity of VM and active allocated task count and size. The improved throttled scheduler will select that VM whose priority is highest among the available set of VM. A priority threshold level is also set to avoid overloading. If the priority of VM is less than priority threshold level, then the task is not allocated to that VM. Also, the scheduler will start searching VM in VM allocation table from the next to the last allocated VM. This will maintain the randomness in task VM mapping. The scheduler will maintain VM allocation table which will store VM id, VM capacity, Active task count, Status and Priority of VM.

2. RELATEDWORKS

Several researchers made proposals on different algorithms load balancing and job scheduling in cloud computing in this section we gave a review of a number of researches that worked to Improve on Load Balancing algorithm.

DOI: 10.31695/IJASRE.2019.33430

[9] attempted to design an efficient scheduling algorithm that uniformly distributes 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 and ESCE algorithms. Throttled 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. the drawback of this algorithm is that overall data center processing time is not much improved in comparison to Equally Spread Current Execution Load(ESCE).

[18] worked on a Hybrid approach having concepts from Round Robin algorithm and Throttled algorithm proposed for virtual machine level load balancing. In the Proposed hybrid approach, initially when a cloudlet is received for execution on VMs then an arbitrary VM is chosen and checked for availability. If it is available then cloudlet is allocated to it. Otherwise control keeps moving in circular way through the VM list until it gets an available VM for cloudlet allocation. After allocation of cloudlet, next comparison takes place at the next VM coming in that circular way. the concept of circular way to allocate VMs to cloudlets has been taken from Round Robin algorithm and inspiration of checking availability on each step has been taken from throttled algorithm .The short coming is that Cost could not be decreased .

[11] proposed an algorithm that combines the methodology of Divide-and-conquer and Throttled Algorithm (DCBT) which schedules the incoming requests to available VMs efficiently and ensures that there is no starvation of the requests. This hybrid approach consists of two algorithms. According to this algorithm requests from different clients are provided to the available Request Handlers (RH) and Virtual Machines (VM). In the initial step, the algorithm checks for the availability of RH's and VM's and divides the requests accordingly using divide and conquer approach. In the next step, the incoming requests are assigned to the different RH's and VM's. Load Balancer keeps track of the current status of each RH or VM and verifies that the current request should only be assigned to the RH and VM which has not been used recently. The algorithm ensures that the load is distributed in an optimized way and no resource is idle thus leading to maximum resource utilization and minimum execution time thus leading to high performance. the drawback is that Deadline constraints are not considered.

[2] proposed a hybrid approach which combines ant colony and particle swarm optimization (ACOPS) to solve scheduling in Virtual machines so that VM's are assigned to servers and resource usage is best utilized. ACOPS uses historical information to predict the workload of new input requests to adapt to dynamic environments without additional task information. Every time the request arrives, the initial step of ACOPS is Pre-reject in which the algorithm checks for the remaining memory of each server and will find the maximum amount of remaining memory. When the memory demands of request exceed the maximum remaining memory, the request will be rejected before scheduling. Search operator is used to construct the solutions for all ants. Practical Swamp Optimization (PSO) operator is applied to improve the search result. In the next step, the evaluation operator is used to estimate the scheduling score of each ant. It will find the best solution and update the global best solution. In the next step, global pheromone updating is applied. This algorithm can serve requests for CPU, memory and disk utilization. Due to pre-reject step, computation time is reduced. Practical Swamp Optimization (PSO) operator further reduces the computing time and improves the scheduling result. Thus, ACOPS ensures better load balancing than individual ant colony and particle swarm optimization algorithms. the drawback is that Although the algorithm can provide high load balancing, make span is shorter in single scheduling.

[12] proposed an Enhanced Hybrid Approach which is the advancement of hybrid algorithm that contains both Throttled and Equally Spread Current Execution algorithm. Enhanced Hybrid algorithm maintains an index list of VM allocation status as well as list to count the allocated request. The allocated request list is compared with the VMs index list. If VMs index list is greater than allocated request list it means that VMs are available to take request else request has been queued until VM is been available. If the VM has been queued, it has to wait in the queue itself. So new host has been created using host create function. In case of availability of VM, the jobs are allocated to that particular VM. And both the index list and hash list are updated. The job in queue needs not wait for long time for the virtual machine to become available. It Maximizes the resource utilization than the existing system. Minimize the response time and negligible idle time. it only analyzes the following parameters namely, the overall response time, data processing time as well as data transfer cost not putting other performance parameters into consideration like the physical memory, storage, band width etc.

3. METHODOLOGY

3.1 Hybrid Load Balance Algorithm

DOI: 10.31695/IJASRE.2019.33430

In this work we developed a hybrid load balancing algorithm based on Equally Spread Current Execution Algorithm ESCE by [3]. which equally distributes loads to virtual machines and improved throttled algorithms by [6]. the distribution of loads to virtual machines is done according to VM's priority and priority threshold value. we followed the steps below to achieve the work .

At first VMs are distributed over hosts according to the host qualifications. The largest number of VMs is located at the most qualified host depending on the Hosts' CPU and memory capacity.

step 1: virtual machines will be distributed over hosts according to host qualification .the largest number of virtual machine is located at the most qualified host depending on CPU capacity and memory .

step 2 :priority will be assigned to each virtual machine and calculated based on CPU, Memory, active allocated count and size of the virtual machines.

step 3:the scheduler will select that VM whose priority is highest and priority threshold is set for virtual machines.

Step 4:if priority of virtual machine is less or equal to priority of threshold level search for overloaded and under loaded virtual machines.

Step 5:if virtual machine is over loaded task is not allocated to that VM rather allocated to next VM and distribute some of the initial VM's work to other VMs with least work so that every VM is equally loaded.

step 6:the scheduler will search for allocation of virtual machine that is next and last on the virtual machine allocation table and store The virtual machines id, memory capacity, active task count ,status & priority.

Figure 1 below shows the working description of the proposed system.

3.2 Conceptual Frame Work Of The Proposed System.

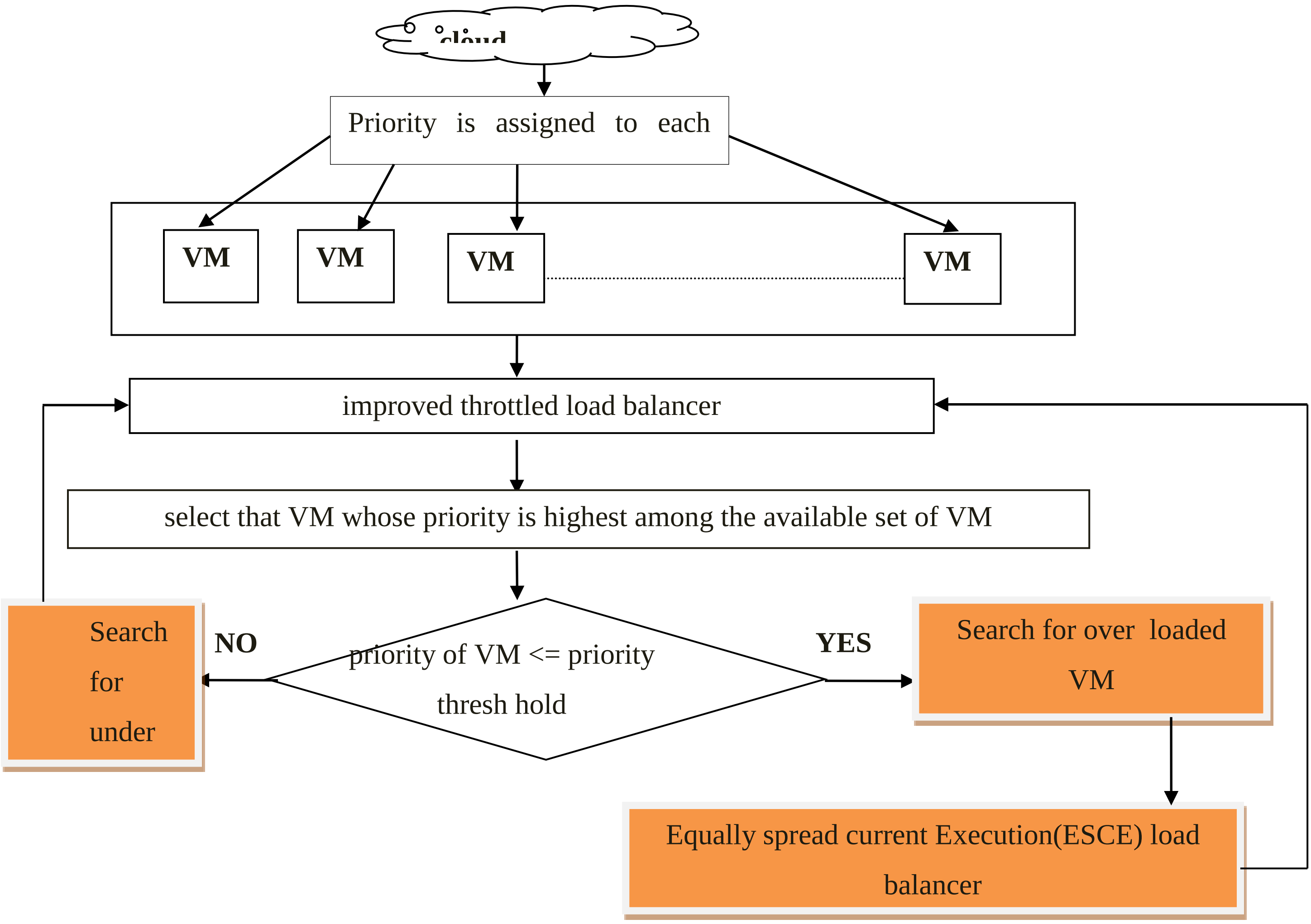

cloud
Priority is assigned to each
VM
VM
VM
VM
improved throttled load balancer
select that VM whose priority is highest among the available set of VM
priority of VM <= priority thresh hold
NO
YES
Search for under
Search for over loaded VM
Equally spread current Execution(ESCE) load balancer

Figure 1.Architecture of the proposed system Proposed in [20].

3.3 Hybrid Algorithm

DOI: 10.31695/IJASRE.2019.33430

The Hybrid Algorithm uses Improved Throttled and Equally Spread current Execution Algorithms (ESCE) for load distribution to all virtual machines so as to get better performance in a heterogeneous environment of hosts. The pseudo code in Figure 2 takes care of the hybrid algorithm, where line 0 to 14 handles the improved Throttled Algorithm by [6] and line 15 to 19 covers the interference of , distributing loads equally when there is over load in Equally Spread Current Execution (ESCE) Algorithm.

```
The Hybrid Algorithm
Input: list of Vms Vm_List, maintain
 an index table of vms with Active task count,
 for every vm table(vm_id),  where k is the
 number of vms selected base on priority.
Output: vm_id is the vm_id that is selected
 for load assignment
0   Initialization:table(0,n-1)=0,k=m,vmid=-1,
vmids=-1,i=0, presentcount=maxvalue,tvmid=-1

1    for i=0 to k// select Vm base on priority
2          tvmid= rand (list)
3          vmid=tvmid
4          if vmid in table(vmid) then
5           presentcount=table(vmid)
6          else
7                      presentcount=0
8          Dmids=(vmid,presentcount)
9    end for
10   tvmid=-1
11   presentcount=0
12   for i=1 to k
13         tvmid=i
14         presentcount=Dmids(tvmid)
15         if presentcount<leastcount then
16                     leastcount=presentcount
17                     vmid=tvmid
18         end if
19   end for
```

Figure 2.The Hybrid Algorithm.

4. EXPERIMENT AND RESULTS

This section gave account of the experiments and results. Cloud Analyst simulator was used to make comparison between the proposed hybrid algorithm and the current load balance algorithms. The Cloud Analyst is a 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 [2] Development of large-scale applications in the cloud using the simulator is very economical and easy. Cloud Analyst is actually made to evaluate performance and cost of large-scale geographically distributed cloud system that is having large user workload based on different parameters. Its GUI is attractive and has huge flexibility to configure any geographical distributed system like setting the hardware parameters (storage, main memory, bandwidth limit, network delays etc.) of a virtual machine or data center. [1].The simulator parameters were identified such as user base configuration, Data centers configuration and VMs configuration. We implemented the hybrid algorithm and the following current load balance algorithms such as Improved Throttled and Equally spread current Execution (ESCE) algorithms.

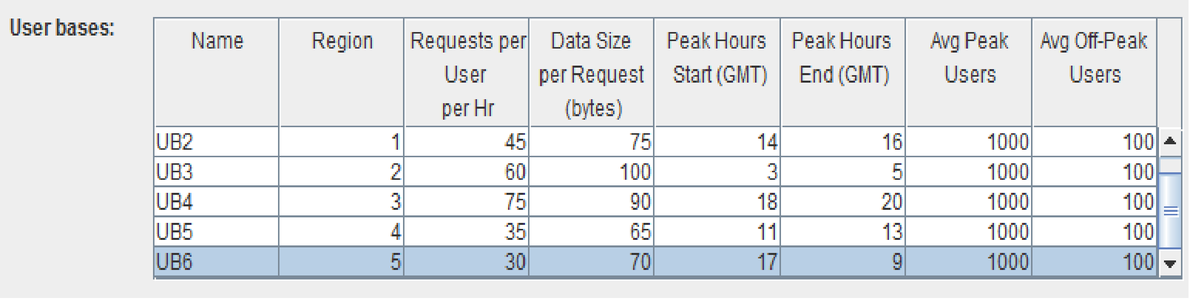
4.1 Experimental Setup

To evaluate the proposed algorithm which considers CPU capacity and physical memory. the experiments were run in a heterogeneous environment of hosts, due to the interest of obtaining accurate better performance we decided to use some of the experimental set up values used by [6]. where each machine has different number of CPUs and varied physical memory with Simulation Duration of 5 Hours, Virtual machine usage cost per hour = $ 1 Memory / second = $ .5 ,Data storage / second = $ 0.25, Data transfer cost / 1GB = $ 0.125, No of processors on physical machine= 3, Processing power= 100 MIPS, Storage devices= 50 GB ,Memory= varied sizes, Internal bandwidth= 1000 MBPS,**VM:** Time-shared**,** Service Broker Policy **is** Optimize Response Time, Number of parallel users from a single user base= 100, No of simultaneous demands a different application server instance can bear= 25, Size of executable instruction per request= 100 bytes ,RAM = 256 MB, Storage quota= 3 GB ,Architecture = x86 ,Operating system = Linux Virtualization technique = VMware, and Bandwidth = 250 MBPS.

4.1.1 User Base Configuration

Table 1 below describs the user base configuration ,taking care of the request per user per hour,Region,Data size per request,peak hours start and end and avg peak users.

Table 1. User Base configuration



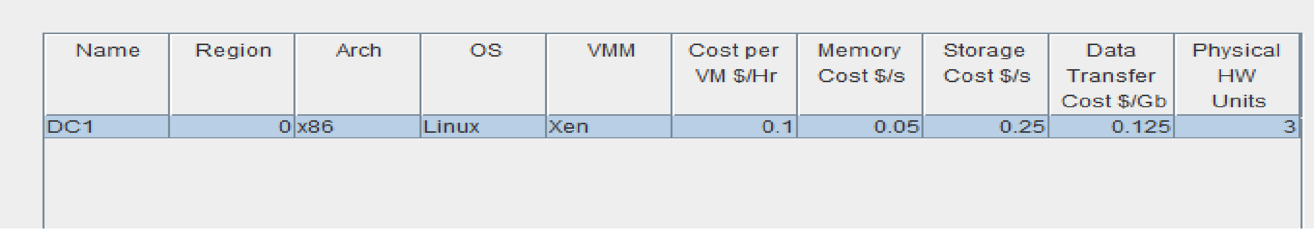
User bases:

Name	Region	Requests per User per Hr	Data Size per Request (bytes)	Peak Hours Start (GMT)	Peak Hours End (GMT)	Avg Peak Users	Avg Off-Peak Users
UB2	1	45	75	14	16	1000	100
UB3	2	60	100	3	5	1000	100
UB4	3	75	90	18	20	1000	100
UB5	4	35	65	11	13	1000	100
UB6	5	30	70	17	9	1000	100

4.1.2 Data Center Configuration Of A Single Data Center

The Table 2 below describes the data centre configuration which includes the Name, Region, Architecture, OS, VM model, cost per VM and memory cost.

Table 2. Data Center configuration

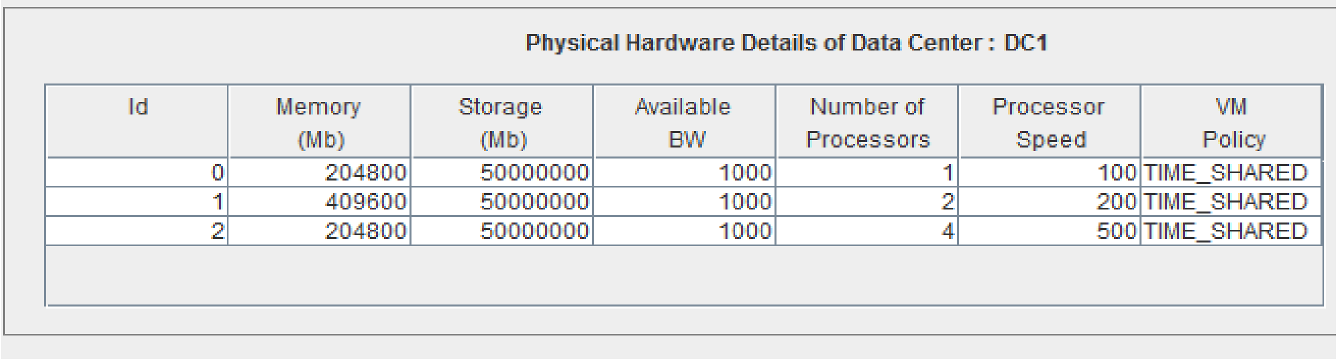


Name	Region	Arch	OS	VMM	Cost per VM $/Hr	Memory Cost $/s	Storage Cost $/s	Data Transfer Cost $/Gb	Physical HW Units
DC1	0	x86	Linux	Xen	0.1	0.05	0.25	0.125	3

4.1.3 Physical Hardware Details Of A Single Data Center

Table 3 below describes the physical hardware configuration which takes care of the Memory, Storage, bandwidth, number of processors and processor speed.

Table3. physical hardware details of a single data center



Physical Hardware Details of Data Center : DC1

Id	Memory (Mb)	Storage (Mb)	Available BW	Number of Processors	Processor Speed	VM Policy
0	204800	50000000	1000	1	100	TIME_SHARED
1	409600	50000000	1000	2	200	TIME_SHARED
2	204800	50000000	1000	4	500	TIME_SHARED

4.2 RESULTS

4.2.1 Results For Existing And Developed System For Testing The Effect Of Network Delay In 5 Scenario.

The Table 4 below shows the overall results obtained for testing the effect of network delay in scenario 1-5 for both the existing Improved Throttled ,Equally Spread Current Execution Algorithm and the proposed Hybrid Algorithm.

DOI: 10.31695/IJASRE.2019.33430

Table4. Results Of Existing And Developed System For Testing The Effect Of Network Delay

scenario		ALGORITHMS		

	AVG(ms)
	G.C($)
		Improved Throttled	ESCE	Hybrid

1 DC 100VM	RT	325.43	355.86	312.83
	PT	19.88	47.14	8.66
	GC	4.15	4.15	2.01
2 DC 50 VM	RT	320.07	365.34	315.61
	PT	15.15	57.47	7.63
	GC	5.65	5.65	3.53
3 DC 35 VM	RT	318.16	381.46	310.40
	PT	13.98	74.92	10.58
	GC	7.15	7.15	5.04
4 DC 25 VM	RT	313.99	374.25	313.68
	PT	10.26	71.40	8.67
	GC	8.66	8.66	6.54
5 DC 20 VM	RT	313.62	344.05	311.90
	PT	10.36	25.94	9.43
	GC	10.16	10.16	8.5

The various graphs derived are displayed in figure 3-9.

4.2.2 **Test Case 1 :**Testing The Effect Of Network Delay Which Consist Of Five Scenarios.

1. Average response time in test case 1 for scenario 1-5

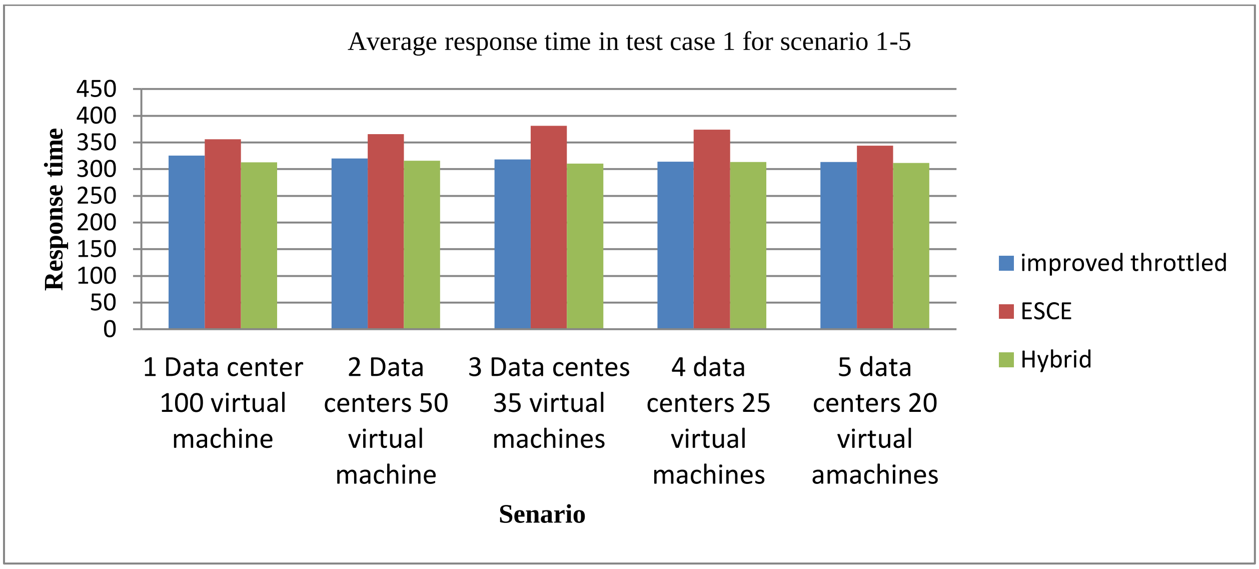


Figure 3. average response time in test case 1 for scenario 1-5

2. Average processing time for test case 1 in scenario 1-5

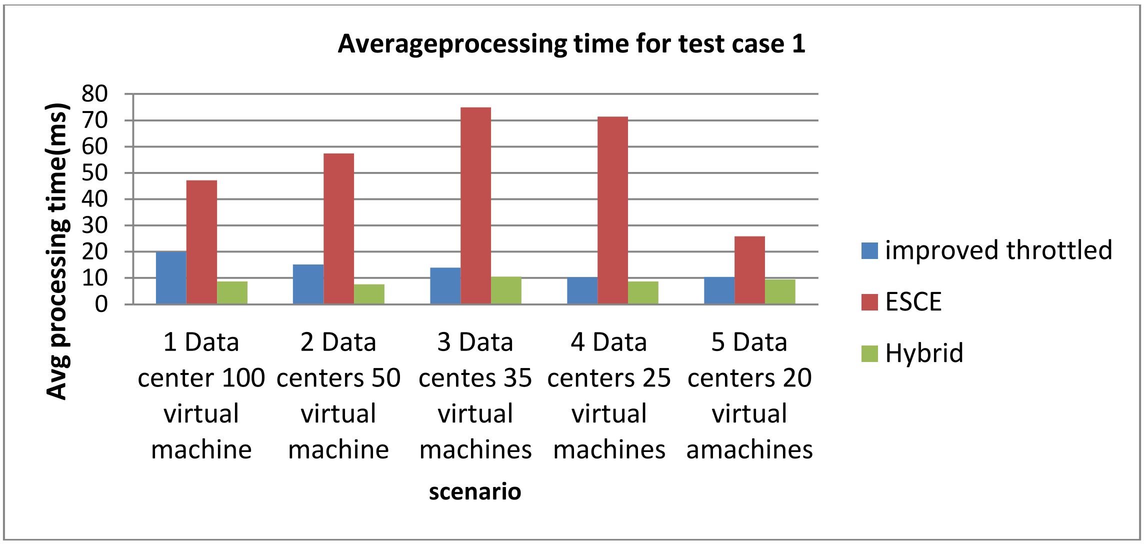


Figure 4. average processing time for test case 1 in scenario 1-5

3. Cost analysis in test case 1 of 5 scenario

DOI: 10.31695/IJASRE.2019.33430

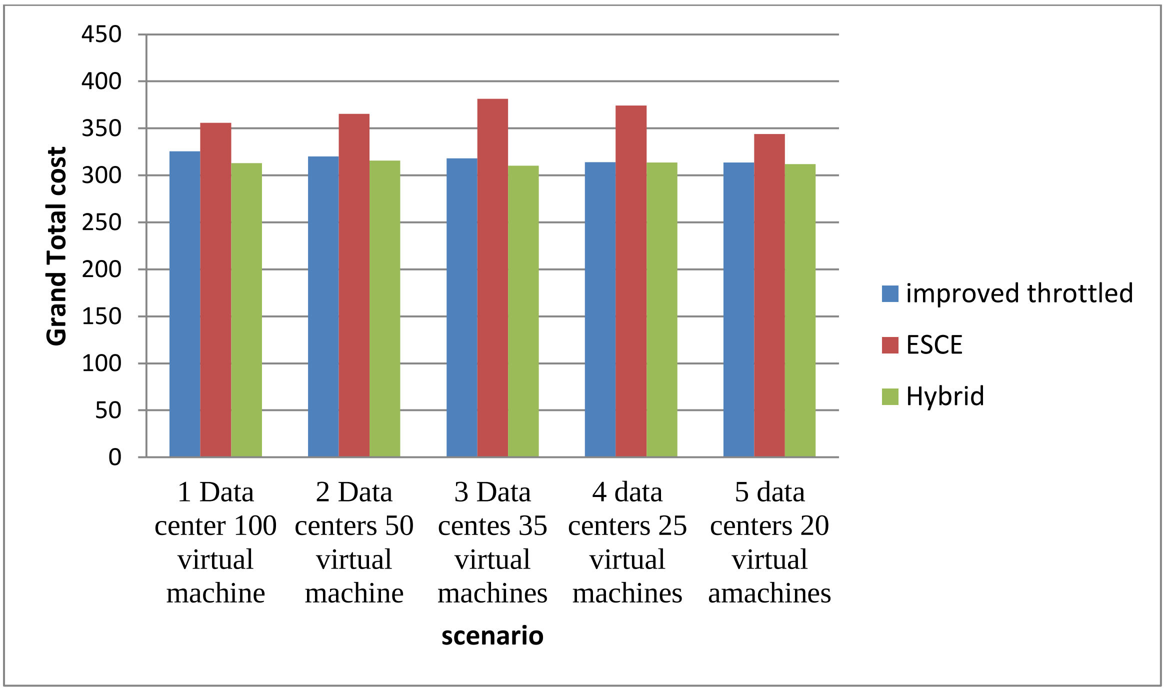


Figure 5. Cost analysis in test case 1 of 5 scenario.

4.2.3 Test case 2 :Testing the effect of physical memory with varied CPU capacity.

1. Response time ,processing time for existing and developed hybrid algorithms for high and varied physical memory.
2.

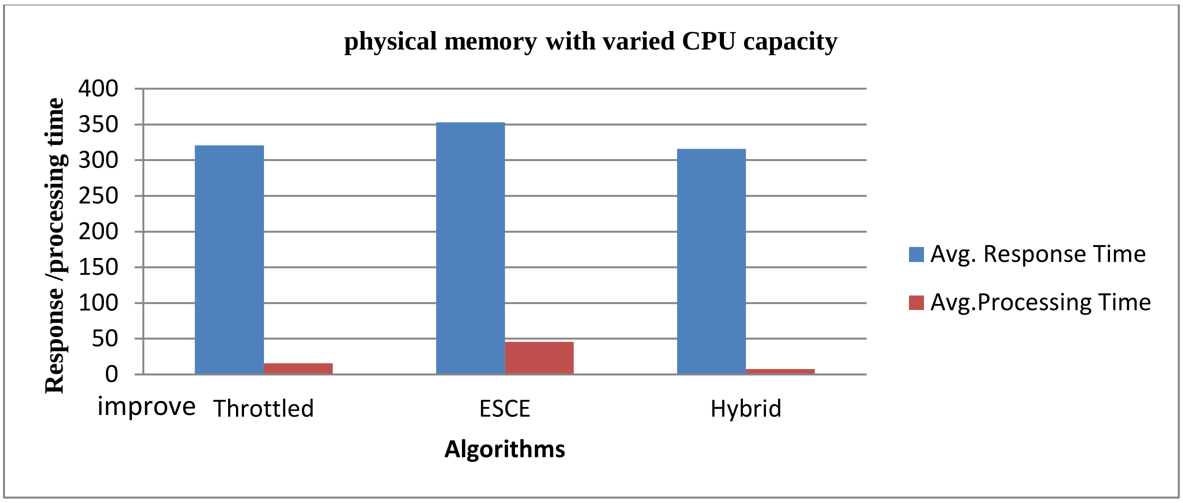


Figure 6.Response time and processing time for high and varied physical memory .

3. Cost analysis for varied physical memory

DOI: 10.31695/IJASRE.2019.33430

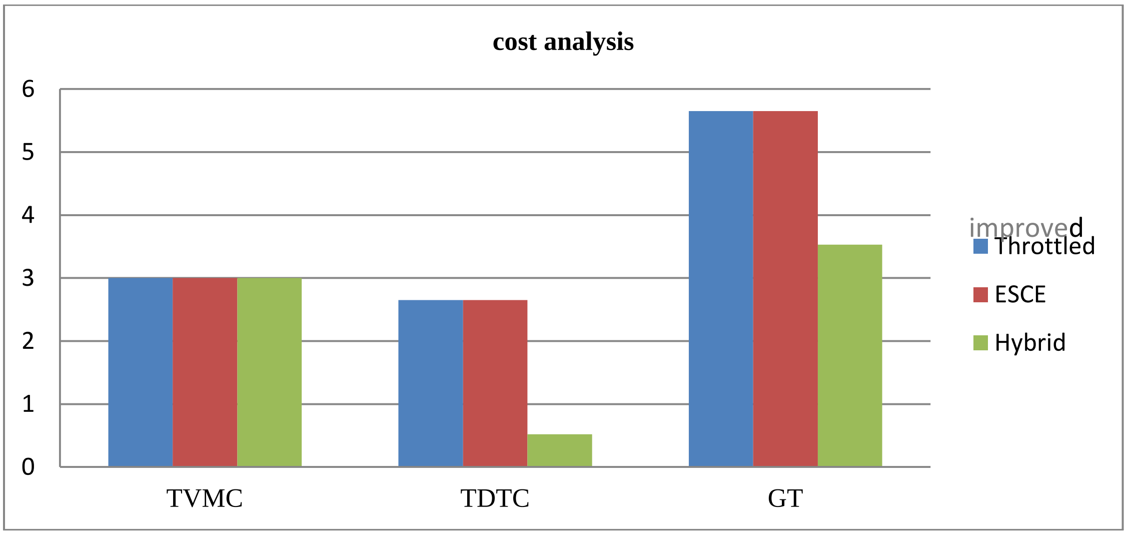


Figure 7: cost analysis for varied physical memory.

4.2.3 Test case 3: Testing the effect of Storage capacity.

1. Response time and processing time in both existing and proposed hybrid algorithms considering storage capacity.

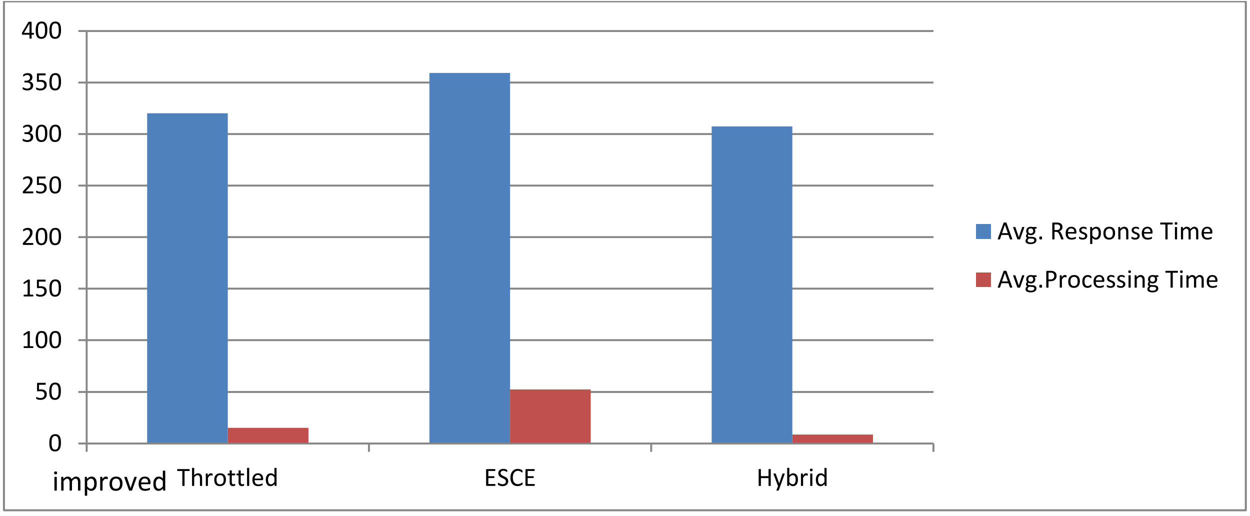


Figure 8: Response time and processing time for storage capacity.

2. Cost analysis for both existing and proposed hybrid algorithms considering storage capacity.

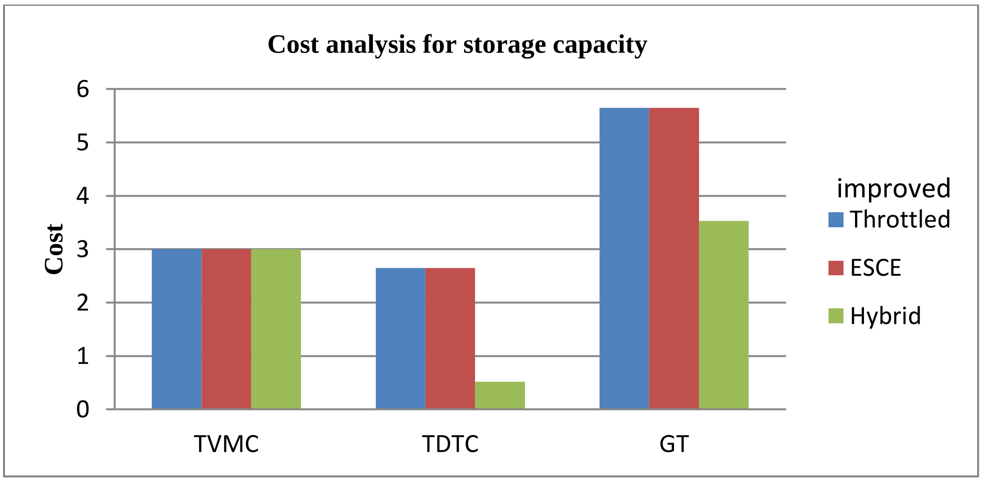


Figure 9: Cost analysis for storage capacity.

DOI: 10.31695/IJASRE.2019.33430

5. DISCUSSION

In Test case 1 all the scenarios shows that the network delay had no effect on the hybrid algorithm which makes it performs better in almost all aspects and it appears to be very good in terms of maximum times due to varied CPU capacity and increased number of RAM. Unlike the existing work done by (imtiyaz *etal* ,2017).

In terms of the physical memory, the hybrid algorithm recorded the best response time of 315.68(ms) and processing time of 7.57(ms). The difference between the processing time results and other algorithms results exceeded 38.00 (ms) on each average processing time and also had the better Max processing time of 33.55(ms).The hybrid also recoded better cost in comparison to other algorithms where the grand total cost was $3.53.

Considering the storage capacity the hybrid had better Avg. response time 307.41(ms) with low storage size .but with high storage it is said to have better Avg. processing time 8.48 (ms) and better max. Response time 577.52 (ms).Also we found that the hybrid algorithm is better than the ESCE and throttled in terms of total cost which is $3.55 in the high storage size with at least a difference of $2.10 each (Table 29). In addition the Throttled recorded the best min. response time 40.30 (ms) and hybrid had the best max. processing time 28.05 (ms) all in high storage size.

6.CONCLUSION AND FUTURE WORK

There has been an increased demand and use of cloud resources by various users all over the world which has drawn so much attention to cloud service providers to provide better load balancing algorithms so as to yield better performances in terms of resource utilization . The use of hybrid algorithm in this research work indicates that combining ESCE and improved throttled algorithms provides better response time, processing time and total cost and also the work considered the effect of resource utilization like the physical memory and storage capacity. in maintaining better performance. In future, fault tolerance and other performances such as Bandwidth, usability etc are to be analyzed so as to ensure a better performance in load balancing and cloud computing in general.

REFERENCES

[1] Arif, A., and Abadhan, S.(2014). Cloud computing simulators: A detailed survey and future direction. *International Advance Computing Conference*-IACC. pp 866-872.

[2] Cho, K., Tsai, P., Tsai C. ,(2014). A hybrid meta-heuristic algorithm for VM scheduling with load balancing in cloud computing Neural Computing and Applications. 26(6), pp 1297-1309.

[3] Bhathiya, .W (2009). Cloud Analyst: A CloudSim-based Tool for Modelling and Analysis of Large Scale Cloud Computing Environments." MEDC Project Report. pp 433-659.

[4] PushpaLatha, K., Dr. Shaji, R. and Jayan, .J. (2014).A Cost Effective Load Balancing Scheme for Better Resource Utilization in Cloud Computing' Journal Of Emerging Technologies In Web Intelligence.6(3), pp 280-290

[5] Hafiz J.and Alaa E.(2015) 'Efficient Load Balancing Algorithm in Cloud Computing' International Journal of Soft Computing and Engineering , 5(3), pp 2231-2307.

[6] Imtiyaz, A., shakeel. A. & souurav, M. (2017). An enhanced load balancing throttled approach for cloud environment. International Research Journal of Engineering and Technology.4(6), pp 512-517

[7] Klaithem A., Nader M., Mariam A., and Jameela, A.(2014). A Novel Approach for Dual-Direction Load Balancing and Storage Optimization in Cloud Services ,IEEE 13th International Symposium on Network Computing and Applications.

[8] Kalai S., Mary, L. and Immaculate, S. (2014). A Survey of Load Balancing Algorithms using VM'B *International Journal of Advancements in Research & Technology*, 3(8), pp 68-76

[9] Mamta K., Sanjay T. (2016) An efficient algorithm for load balancing in cloud computing *International Journal Of Engineering Sciences & Research Technology*.5(6), pp 468-475

[10]Sasikala, N. & Ramesh, D. (2014).Effective Load Balancing for Cloud Computing using Hybrid AB Algorithm .2(4), pp 3815-3822.

DOI: 10.31695/IJASRE.2019.33430

[11] Shridhar, G. and Ram, M. (2013).Load Balancing in Cloud Computing Using Modified Throttled Algorithm ,IEEE International Conference on Cloud Computing in Emerging Markets (CCEM).

[12] Subalakshmi, S., Malarvizhi, N. (2017).Enhanced Hybrid Approach for Load Balancing Algorithms in Cloud Computing. *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*.2(2), pp 136-142

[13] Radha R. (2015). An Approach to Load Balancing In Cloud Computing, Copyright to IJIRSET *4(5),3769-3777.*

[14] Padhy, R.P. & Rao, G.P. (2011) Load Balancing in Cloud Computing Systems: Orissa, India,.

[15] Ray, S. and DeSarkar, A.(2012). Execution analysis of load balancing algorithms in cloud computing environment**.** *International Journal on Cloud Computing: Services and Architecture. 2(5), 1-13*.

[16] Bhathiya, W. and Roderigo, N.(2010). Cloud Analyst: A Cloud-Sim-Based Visual Modeler For Analyzing Cloud Computing Environments And Applications. *Proc Of IEEE International Conference On Advance Information Networking And Applications.*

[17] [29] Padhy, R.P. & Rao, G.P. (2011) Load Balancing in Cloud Computing Systems: Orissa, India,.

[18] Rajkumar, S. & Jyotsana O.(2014). A Hybrid Approach for VM Load Balancing in Cloud Using CloudSim *International Journal of Science, Engineering and Technology Research*.3(6) ,1734-1739.

[19] Bhathiya, W. and Roderigo, N.(2010). Cloud Analyst: A Cloud-Sim-Based Visual Modeler For Analyzing Cloud Computing Environments And Applications. *Proc Of IEEE International Conference On Advance Information Networking And Applications.*

[20] Boukari, S.& Abdullahi, N. (2019).A Hybrid Approach To Enhance Load Balancing In A Heterogeneous Cloud Environment *International Journal Of Advanced Research In Computer And CommunicatioEngineering*.8(4).