ISSN: 2277-9655
Impact Factor: 4.116

IJESRT

INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

AN EFFICIENT ALGORITHM FOR LOAD BALANCING IN CLOUD COMPUTING

Mamta Khanchi*, Sanjay Tyagi

* Research Scholar, Department of Computer Science and Applications, Kurukshetra University, Kurukshetra-136119

Assistant Professor, Department of Computer Science and Applications, Kurukshetra University, Kurukshetra-136119

DOI:

ABSTRACT

Cloud computing is a computing provided over the internet. The principle aspect of cloud computing is virtualization that deals with the construction and management of virtual machines efficiently. As the number of consumers and requests for the services are increasing day by day in cloud computing, therefore load balancing is an important research area for handling the users' requests efficiently. For efficient & effective management and usage of cloud service provider's resources, many load balancing algorithms have been already proposed. This paper proposed and implemented a hybrid approach for virtual machine level load balancing.

KEYWORDS: Cloud Computing, Load Balance, Virtualization, Virtual Machine.

INTRODUCTION

Cloud computing promotes the provision and use of IT infrastructure, platform and applications of any kind in the form of services that are electronically accessible via internet in a dynamically scalable and metered manner. It is a prominent research field accepted in academic as well as industrial world[1]. Three type of services provided over the cloud are:-Software as a service (SaaS), Platform as a service (PaaS) and Infrastructure as a services (IaaS). Based on the architectural model, cloud computing can be divided in to public, private, hybrid and community clouds.

Virtualization is the principle aspect of cloud system. Using virtualization technique, cloud computing virtualized a single system into number of virtual systems. Basically a virtual machine is a software implementation of physical resource [2]. A hypervisor (a firmware or low-level program) also called virtual machine monitor is responsible for sharing of a single physical instance of cloud resources among various tenants [3]. Relying on the layer where virtualization takes place, two major categories of virtualization can be described: Container-based and Hypervisor-based virtualization.Container-based virtualization, also called jail virtualization takes place at the operating system level, while hypervisor-based virtualization is accomplish at the hardware level.

Efficient resource usage, server consolidation and energy conservation and less space requirement are the major benefits of vitualization technology. Instant access to the data is the main drawback of virtualization, as it does not provide the sufficient data protection.

LOAD BALANCING IN CLOUD ENVIRONMENT

The cloud system architecture is principly parallel and distributed in nature. This distributed architecture set up the resources distributively for delievering the services to cloud consumers, located in different geographical areas. The cloud users randomly make cloudlet request in distrbuted environment. These requests may unevenly distributed over the processors due to randomness. This uneven cloudlet allotment may result in imbalance i.e. some processors may get overloaded while other may remain underloaded. Therefore there is requirement of load balancing. The load balancing distributes workload among multiple computing resources such as cores of processors or disk drives. In VM scheduling, the major objective of workload balance is to transfer the load from overloaded virtual macines to

underloaded virtual machines. Additionally, other objectives are to optimize resource usage, maximize throughput and minimize response time. Two types of load balancing algorithms are: static and dynamic. Static load balancing algorithms are simple to design and takes very less execution time. These algorithms works on the basis of prior knowledge of tasks and resources of system. But these algorithms can only be applied in the scenario, where all the hosts in the cluster have same processing capability. Examples are Round Robin, weighted round robin, least connection scheduling algorithm and many more. Dynamic load balancing algorithms are more flexible, reliable and capable of handling large number of users requests than static algorithms. These algorithms rely on the present state of the sytem and are best suited for changing environment. These are self adaptive algorithms and deals with the cloudlet requests that generates different workloads, which are difficult to predict.

Existing load balancing algorithms

To design an effective load balancing policy and to determine how to increase the cloud resouce usage are the two main goals of a cloud service provider. The VM scheduling algorithms for laod 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:

RoundRobinVM Load Balancing Algorithm

It is a very simple load balancing algorithm that places the newly coming cloudlets on the available virtual machines in a circular manner. The major advantages of this algorithms 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.

Throttled VM Load Balancing Algorithm

It is a dynamic approach. In this, user submits its request to the DataCenter Controller(DCC). Data Center Controller asks the VM Load Balancer to determine the appropriate virtual machine that can handle that much workload easily. Throttled VMLoadBalancer 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 VMLoadBalancer accept the cloudlet request and allote the cloudlet request over that virtual machine. Otherwise, client have to wait in the waiting queue untill 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[4].

ESCE VM 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[5]. As the name indicate, it equally distribute the workload on each VM in data center. A job queue keeps all the cloudlet requests that needs the VM for their execution. ESCE VMLoadBalancer (VMLB) also maintains a list of virtual machines. The VMLoadBalancer continously check the job queue and VM list. If a VM is found free, then cloudlet request will be alloted 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 underloaded virtual machine, so as to reduce some load of overloaded VM. The main drawback is high computational overhead[6].

RELATED WORK

Rajkumar Somani, Jyotsana Ojha[7], propsed a hybrid approach for balance of workload among virtual machines. The performance of hybrid approach was found better when compared with the approaches discussed above. Subhashish Mohapatra et al.[6], implemented four algorithms named as RR, Throttled, ESCE, FCFS for balance of load using cloud analyst tool. In this, performance of Round robin was proved to be the best among all other. B. Santhosh Kumar and Latha Parthiban[8], implemented an algorithm for load distribution among VMs. The algorithm was based on the mechansim of least frquently used. The main objective was to keep VM busy all the time for their efficient utilization. The proposed algorithm outperformed the other existing algorithms. Jinhua Hu et al.[9], gave an idea on virtual machine load balancing for efficient utilization of VMs.

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 (AVLB) 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 underloaded 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.

Knowledge of VM States and allocated cloudlets are the two main features of Throttled and Active VM Load Balancer Scheduling algorithms respectively. These features combined together, make the proposed scheduling algorithm more efficient & effective and help in fair distribution of the load.

Hybrid VM Load Balancing Algorithm

Input- Userbases/Cloudlets UB1, UB2,... UBn.

-Available VMs VM1, VM2, VM3,…,VMn within data center.

Step1: Hybrid VM Load Balancer maintains a list of VMs, their states (AVAILABLE/BUSY) and allocated cloudlets list. Initially state of every VM is AVAILABLE and allocated cloudlet list is empty.

Step 2: Data Center Controller gets cloudlet requests from cloud clients.

Step 3: Data Center Controller asks the Hybrid VM load Balancer for available VM.

Step 4: Hybrid VM load Balancer do

a) Find the next available VM using VM State List.
b) Check if present allocation count is less than maximum VM list length and length of virtual machine list is greater than zero, then allocate the VM.
c) Determine the current cloudlet load on every VM.
d) Return vmId of those VM which have minimum load.

Step 5: Hybrid VM Load Balancer allocates the cloudlet over available VM.

Step 6: If a VM get overloaded then hybrid VM Load balancer moves some workload on VM that have minimum workload

Step 7: The DCC get the reply of sent cloudlet & then allots a waiting request from job pool to hybrid VM Load Balancer.

Step 8: continue with step 4.

Output- Userbases/cloulets are allocated on the available VMs and completed with minimum response time and processing time at DC.

EXPERIMENTAL SETUP

In this study, the proposed VM Load Balancing algorithm is implemented using the following softwares and tools: windows7 operating system, NetBeans IDE 8.1, JDK 1.8 and cloud analyst tool. This algorithm is implemented for IaaS (Infrastructures as a Service) model in a simulated cloud environment.

Cloud Analyst

Cloud Analyst is an extension to cloudsim oriented simulator used for modeling and simulation of real cloud environment [10]. Some key elements of Cloud Analyst simulator are as follow:

Region: World is split into six regions in Cloud Analyst, that symbolize the six continents (Asia, Australia, Africa, North America, South America and Europe). User base and datacenter which are very essential components of cloud analyst, resides in these regions.

UserBase: A group of users are modeled using this component which is taken as a single unit. Traffic generation is the major responsibility of this component.

Data Center Controller (DCC): It is the most important component in cloud analyst. A single cloudsim maps a single DCC. Various activities like cloudlet request routing, creation and destruction of virtual machines are managed using DCC.
VM Load Balancer: VM load Balancer is used to decide the assignment of cloudlets on VMs.
Cloud Application Service Broker: Traffic routing is managed between user bases & data centers using this element.

Simulation Parameters
In addition to existing load balancing policies such as round robin, throttled and ESCE, a new policy named as Least Response Time VM Load Balancing is added to cloud analyst tool as shown in figure 1.

For algorithm simulation, six user bases named as UB1, UB2, UB3, UB4, UB5, UB6 and four data centers named as D1, D2, D3, D4 are created. Data centers D1, D2, D3 and D4 are created in region R0, R4, R2 and R3 respectively. User Bases UB1, UB2, UB3, UB4, UB5 and UB6 are created in R0, R1, R2, R3, R4 and R5 regions respectively as shown in figure 2. User base configurations such as their regions, requests per user per hr, data size of each request, average peak users and so on, are shown in figure 3.

Each data center is constructed with 15 physical machines, each have following configuration: 8086 architecture, linux operating system, Xen virtual machine manager, 204800 RAM (MB), 100TB storage space, 1000000 available bandwidth, 4 processors, 1000 MIPS , TIME_SHARED VM scheduling policy as shown in figure 4 & 5 . Each physical machine is consolidated with 10 virtual machines as shown in figure 6. Each virtual machine is configured as follow: 1000 MB Image size, 512 MB memory, 1000 MB bandwidth.

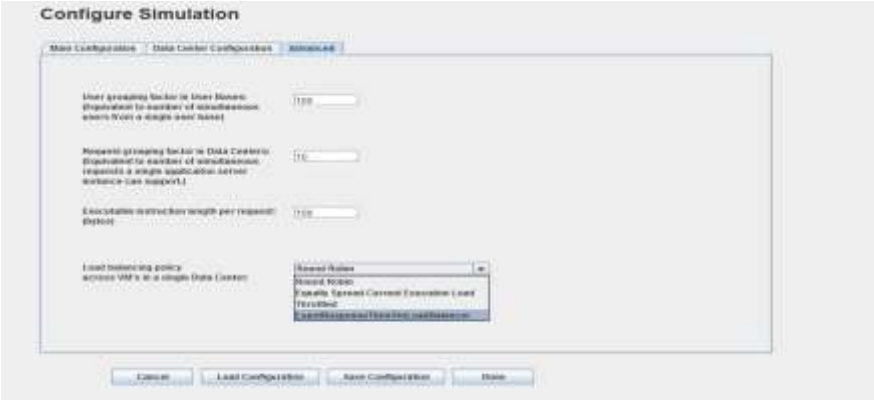

Fig 1: A new policy Least Response Time VM Load Balancing is added to cloud analyst.

Fig 2: Illustration of different regions and data centers on the globe.

ISSN: 2277-9655
Impact Factor: 4.116

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
UB1	0	60	1000	3	9	10000	100
UB2	1	60	1000	3	9	10000	100
UB3	2	60	1000	3	9	10000	100
UB4	3	60	1000	3	9	10000	100
UB5	4	60	1000	3	9	10000	100

Fig 3: Configuration of User bases

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	0.1	0.1	15
DC2	4	x86	Linux	Xen	0.1	0.05	0.1	0.1	15
DC3	2	x86	Linux	Xen	0.1	0.05	0.1	0.1	15
DC4	3	x86	Linux	Xen	0.1	0.05	0.1	0.1	15

Fig 4: configuration of data centers

Physical Hardware Details of Data Center : DC1

Id	Memory (Mb)	Storage (Mb)	Available BW	Number of Processors	Processor Speed	VM Policy
0	204800	100000000	1000000	4	10000	TIME_SHARED
1	204800	100000000	1000000	4	10000	TIME_SHARED
2	204800	100000000	1000000	4	10000	TIME_SHARED
3	204800	100000000	1000000	4	10000	TIME_SHARED
4	204800	100000000	1000000	4	10000	TIME_SHARED

Fig 5: Configuration of physical hardware machine

Application Deployment Configuration:

Service Broker Policy: Optimise Response Time

Data Center	# VMs	Image Size	Memory	BW
DC1	10	10000	512	1000
DC2	10	10000	512	1000
DC3	10	10000	512	1000
DC4	10	10000	512	1000

Fig 6: Application Deployment Configuration with service broker policy and VM parameters

PERFORMANCE ANALYSIS

The parameters/configurations defined above have been used for each VM load balancing scheduling algorithms one by one. Results are calculated for following matrices: average response time by region, request servicing time of data centers, overall response time and overall processing time of data centers as shown in table 1, table 2, table 3 and table

4 respectively. After comparative analysis, it is proved that hybrid load balancing algorithm give better results when compared with three existing VM load balancing algorithms.

Response Time: The amount of time taken between the cloudlet requests sent and response received by cloud consumer is called response time. Overall Response time is calculated using formula:

Overall response time= avg. response time total/count (1)

Here count is the number of user bases created.

Similarly, overall data center processing time is calculated using formula:

Overall data center processing time= data center request servicing time total/count (2)

Here count is the number of data center created.

Table 1: comparison among all load balancing algorithms (Average response time (ms))

User bases	Round Robin	Throttled	ESCE	Hybrid
UB1	55.25	52.23	49.21	44.07
UB2	199.90	196.69	191.25	188.13
UB3	57.11	53.49	49.23	43.12
UB4	56.23	52.37	49.12	42.73
UB5	58.29	54.26	46.23	42.23
UB6	204.49	201.23	198.01	194.39

Table 2: Comparison between request servicing time (ms) of data centers (Among all algorithms)

Data	Round Robin	Throttled	ESCE	Hybrid
DC1	0.68	0.45	0.32	0.25
DC2	0.48	0.44	0.41	0.38
DC3	0.45	0.43	0.32	0.27
DC4	0.57	0.48	0.44	0.34

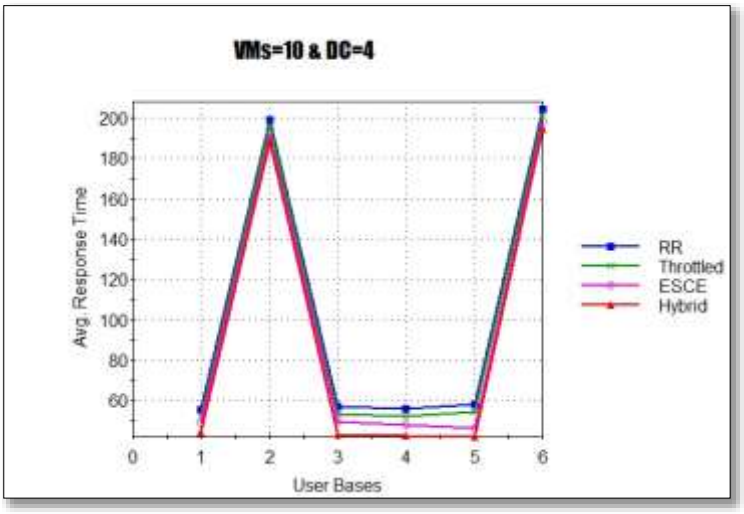

Fig 6: analysis showing six user bases vs. average response time (ms)

Figure 6, shows the graphical representation of the performance of all four algorithms. The Experiment is performed with four data centers and ten virtual machines on each data center configuration. Vertical axis shows the average

response time (in milliseconds), while horizontal axis shows the User Bases. For User Bases UB1, UB2 & UB6, all four algorithms have near/close average response time (however, the proposed hybrid approach has minimum average response time among all.), while for User Bases UB3, UB4 & UB5, the proposed approach provide the least response time.

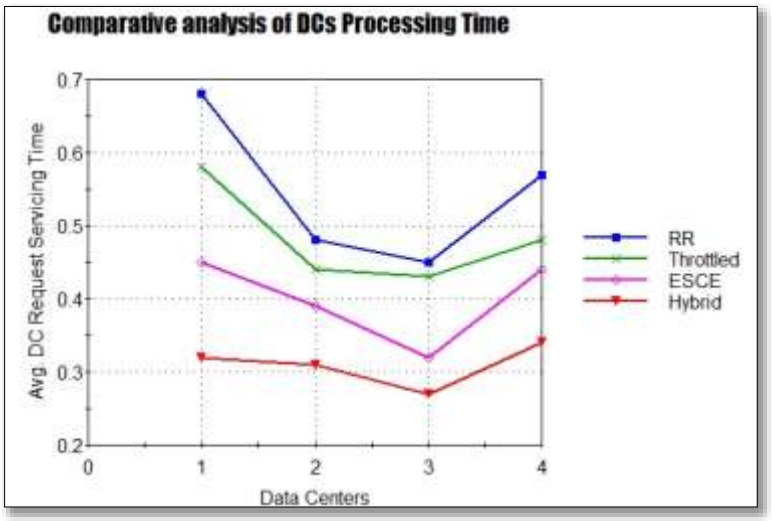

Fig 7: Analysis showing four data centers and their request servicing time (ms)

Figure 7, shows the graphical representation of all four algorithms for average data center request servicing time (in milliseconds). Vertical axis shows the average data center request servicing time (in milliseconds), while the horizontal axis shows the data centers. For each data center, the proposed hybrid algorithms provide the least data center request servicing time.

Table 3, shows the comparison among Round Robin, Throttled, ESCE and hybrid algorithms for overall response time (in milliseconds). The overall response time is calculated using equation 1. Bold values indicate that the proposed hybrid algorithm provide minimum response time.

Table 3: Overall response time for all algorithms

Load Balancing Policies	**RR**	**Throttled**	**ESCE**	**Hybrid**
Overall Response Time(ms)	105.38	101.71	97.175	**92.45**

Table 4: Overall data center processing time for algorithms

Load Balancing Policies	**RR**	**Throttled**	**ESCE**	**Hybrid**
Overall Data Center processing time(ms)	0.55	0.45	0.37	**0.31**

Table 4, shows a comparison of overall data center processing time among Round Robin, Throttled, ESCE and Hybrid algorithms. The overall data center processing time is calculated using equation 2. Bold values indicate that the proposed hybrid approach outperforms the other approaches (i.e. have least overall data center processing time).

CONCLUSION AND FUTURE SCOPE

In present circumstances, Cloud computing is broadly accepted IT-based service. However, there are many issues that have not been completely figured out such as real time scheduling, load balancing, VM migrations and many more. Balance of load is a major issue in cloud that deals with the efficient and scalable distribution of workload. It also takes care that all the computing resources must be fairly/uniformly distributed. The proposed hybrid approach has been implemented using cloud analyst simulation tool and results depict that the proposed approach outperform the

RR, throttled and ESCE algorithms in terms of average and overall response time. Overall data center processing time is not much improved in comparison to ESCE. Thus, this problem can be solved out in future. Also the implementation of proposed hybrid approach can be accomplished in the real cloud environment in the future.

REFERENCES

[1] T. Valte, T. J. Valte and R. Elsenpeter, Cloud Computing - A Practical Approach, TATA McGRAW-HILL, 2010.
[2] S. Angeles, 20 january 2014. [Online]. Available: http://www.businessnewsdaily.com/5791-virtualization-vs-cloud-computing.html. [Accessed 20 april 2016].
[3] J. Celestia, "Cloud Computing Infrastructure," 2016. [Online]. Available: http://www.tutorialspoint.com/cloud_computing/cloud_computing_infrastructure.htm. [Accessed 2016].
[4] S. Nayak and P. Patel, "Analytical Study for Throttled and propsed Throttled algorithm for load Balancing in Cloud Computing using Cloud Analyst," International Journel of Science Technology & Engineering, vol. 1, no. 12, pp. 90-100, 2015.
[5] R. Kumar and T. Prashar, "Performance Analysis of Load Balancing Algorithms in cloud computing," International Journal of Computer Applications, vol. 120, no. 7, pp. 19-27, June , 2015.
[6] S. Mohapatra, K. S. Rekha and S. Mohanty, "A comparision of four popular heurisitcs for Load balancing of Virtual Machines in Cloud Computing," International journal of Computer Applications, vol. 68, no. 6, pp. -38, 2013.
[7] R. Somani and J. Ojha, "A Hybrid Approach for VM Load Balancing in cloud using Cloudsim," International Journal of Science, Engineering and Technology Research (IJSETR), vol. 3, no. 6, pp. 1734-1739, June 2014.
[8] S. B. Kumar and L. Parthiban, "An implementation of load balancing policy for Virtual Machines associated with a data center," Internation Journal of Computer Science and Engineering Technology, vol. 5, no. 3, pp. 253-261, 2014.
[9] J. Gu, J. Hu, T. Zhao and G. Sun, "A scheduling strategy on load balancing balancing of Virtual machine Resourcees for Cloud Computing," in Parallel Arcchtectures, Algorithms and Programming (PAAP), 2010.
[10] B. Wickremasinghe, "CloudAnalyst: A CloudSim-based Tool for Modelling and Analysis of Large Scale Cloud Computing Environments," 2009.
[11] Elektronnmania, "Cloud Computing," 27 june 2013. [Online]. Available: https://www.youtube.com. [Accessed 7 april 2016].
[12] Baun, Cloud Computing, Heidelberg: Springer-Verleg Berlin, 2011.
[13] J. Hu, J. Gu, G. Sun and T. Zhao, "A scheduling Strategy on load balancing of Virtual Machine Resources in Cloud Computing Environment," IEEE, 2010, pp. 89-96.
[14] J. J. Wang and S. Mu, "Security Issues and Countermeasures in cloud computing," in IEEE international conference on Grey Systems and Intelligent service (GSIS), Nanjing, 2011.
[15] "Load Balancing," [Online]. Available: https://en.wikipedia.org/wiki/Load_balancing_(computing). [Accessed 21 april 2016].