

Analytic Study Of Load Balancing Techniques Using Tool Cloud Analyst.

Tanveer Ahmed

University School of Information Technology
M.Tech CSE, GGSIPU, Dwarka
New Delhi, India

Yogendra Singh

University School of Information Technology
M.Tech CSE, GGSIPU, Dwarka
New Delhi, India

Abstract

Cloud computing, as of today has boomed the use of internet services and has enhanced the use of network services where the capability of one node can be utilized by other nodes. Central to the architecture is SoA and IS which raises a big question on performance and complexity. To properly manage the resources of the service provider we require balancing the load of the jobs that are submitted to the service provider. Load balancing is required as we don't want one centralized server's performance to be degraded. A lot of algorithms have been proposed to do this task. In this paper we present a comparison of various policies utilized for load balancing using a tool called cloud analyst.

Keywords - Cloud Computing, Load balancing, Cloud Analyst, CloudSim, Virtual Machine.

1. INTRODUCTION

Cloud computing [1] is service which has allowed users to utilize the potential of other nodes on the internet and then use the same node's services to perform the desired task. It is usually composed of both the services delivered by the service provider and the datacenter that provides the services. Cloud computing provides all the features of grid computing, software as a service and utility computing. It utilizes the concept of virtualization. Cloud computing architectures are inherently parallel, distributed and serve the needs of multiple clients in different scenarios. It allows an organization to focus on paying for the services they require rather than utilizing traditional software services in which a fully fledged system must be built. In [2], cloud computing is thought as of dealing with the three issues, we here in the paper focus on the property of decision. To perform better execution of the operations for the nodes one needs additional attributes for performance calculation. These additional attributes are bound to increase the issue of complexity. The major issues associated with this paradigm are of security, complexity and reliability. Load balancing is the mechanism that decides which requesting nodes/client will use the virtual machine and which requesting machines will be put on hold. Load balancing can be done individually as well as on grouped basis. Load balancing is also required to minimize the cost of machine and maximize the profit for the service being offered. Hence with these issues in mind we here in this paper propose a systematic study of the various algorithms used in cloud

computing that will give a better idea in terms of overall response time, cost, and datacenter service requesting time.
2. algorithms used

2.1 Round Robin

It is one of the simplest scheduling techniques that utilize the principle of time slices. Here the time is divided into multiple slices and each node is given a particular time slice or time interval i.e. it utilizes the principle of time scheduling. Each node is given a quantum and in this quantum the node will perform its operations. The resources of the service provider are provided to the requesting client on the basis of this time slice. Though the algorithm is very simple but there is an additional load on the scheduler to decide the size of quantum. It is shown in fig. 1.

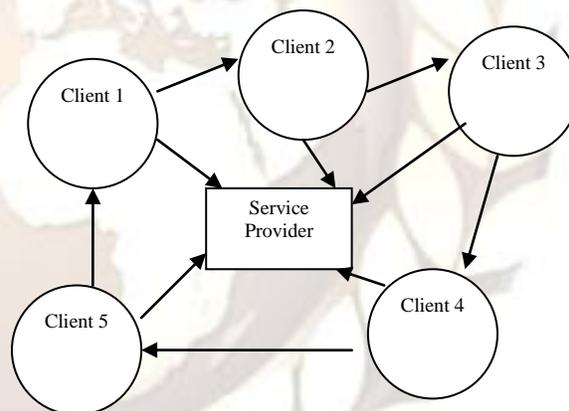


Figure1. Round Robin Algorithm

2.2 Equally spread current execution load

It is spread spectrum technique in which the load balancer spread the load of the job in hand into multiple virtual machines. The load balancer maintains a queue of the jobs that need to use and are currently using the services of the virtual machine. The balancer then continuously scans this queue and the list of virtual machines. If there is a VM available that can handle request of the node/client, the VM is allocated to that request. If however there is a VM that is free and there is another VM that needs to be freed of the load, then the balancer distributes some of the tasks of that VM to the free one so as to reduce the overhead of the former VM. Figure2. better explains the working of the ESCE algorithm. The jobs are submitted to the VM manager, the load also maintains a

list of the jobs, their size and the resources requested. The balancer selects the job that matches the criteria for execution at the present time. Though there algorithm offers better results as shown in further section, it however requires a lot of computational overhead.

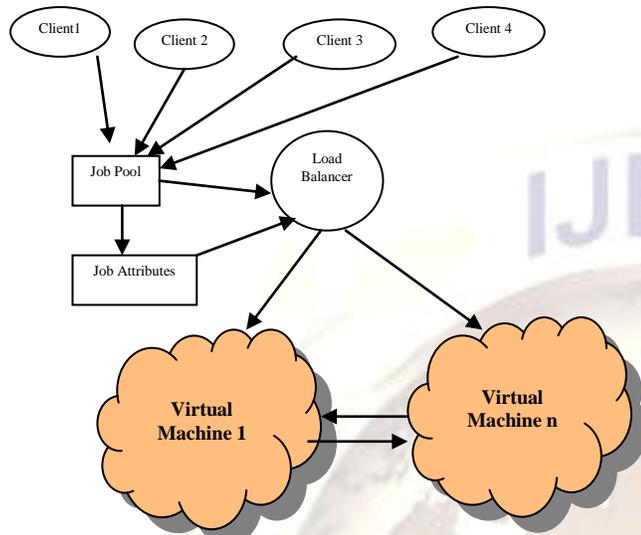


Figure2. ESCE Algorithm

2.3 Throttled Load balancing

In this algorithm the client first requests the load balancer to find a suitable Virtual Machine to perform the required operation. It is shown in fig.3. The process first starts by maintaining a list of all the VMs each row is individually indexed to speed up the lookup process. If a match is found on the basis of size and availability of the machine, then the load balancer accepts the request of the client and allocates that VM to the client. If, however there is no VM available that matches the criteria then the load balancer returns -1 and the request is queued.

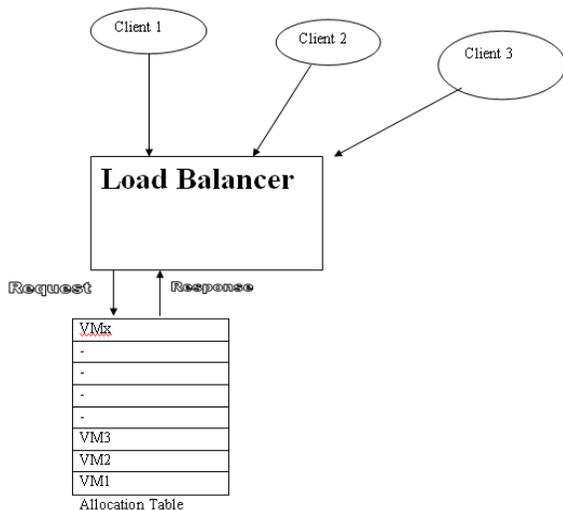


Figure3. Throttled Load Balancing Algorithm

3. CLOUD ANALYST

In order to study the various load balancing techniques we are using cloud analyst in this three techniques are given that are

round robin, equally spread current execution load (ESCEL), and throttled. These load balancing techniques are used by the VmLoadBalancer component of the cloud analyst. These techniques determine the order in which the application must be deployed on the data centers. The default technique used by cloud analyst is Round robin algorithm; this algorithm simply allocates incoming requests in a round robin fashion so this algorithm does not take into account the current load of each virtual machine. The second policy used is equally spread current execution as the name specifies this policy equally distributes the load on each VM. And the third one is throttled, in this policy a threshold is defined for assigning request to each VMs, if the number of requests increase beyond this threshold value in all available virtual machines then the requests are queued until a VM becomes available.

Cloud Analyst[3] is a GUI based tool that is developed on CloudSim[4]architecture[5]. CloudSim is a toolkit that allows doing modeling, simulation and other experimentation. The main problem with CloudSim is that all the work need to be done programmatically. The cloud analyst tool removes all the complexities by making GUI so that focus can be done on simulation rather than programming. It allows the user to do repeated simulations with slight change in parameters very easily and quickly. The cloud analyst allows setting location of users that are generating the application and also the location of the data centers. In this various configuration parameters can be set like number of users, number of request generated per user per hour , number of virtual machines, number of processors, amount of storage, network bandwidth and other necessary parameters. Based on the parameters the tool computes the simulation result and shows them in graphical form. The result includes response time, processing time, cost etc .By performing various simulations operation the cloud provider can determine the best way to allocate resources, based on request which data center to be selected and can optimize cost for providing services. The various activities performed in cloud analyst tool are summarized below

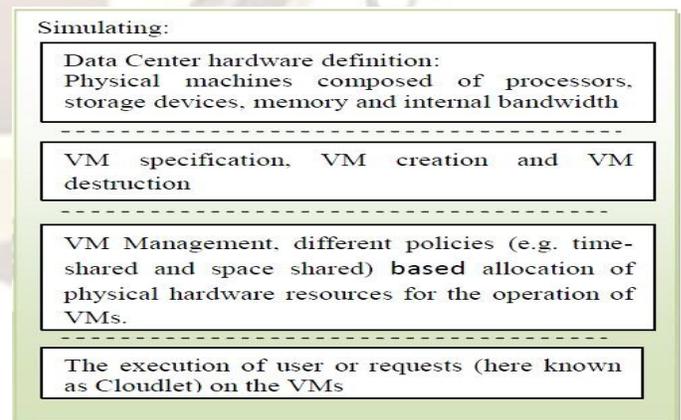


Figure4. Activities done in cloud analyst.

The main components of cloud analyst tool are:

GUI Package. It provides various user interfaces to configure the various simulation parameters in a better and easy way. The GUI of cloud analyst is shown in figure5.



Figure5. GUI of cloud analyst.

Simulation. Based on the various parameters set it executes the simulation to provide necessary results.

UserBase. In this user base is modeled to represent the users who deploy application.

DataCenterController. This component is used to control the various data center activities.

InternetCharacteristics. In this component various internet characteristics are modeled simulation, which includes the amount of latency and bandwidth need to be assigned between regions, the amount of traffic, and current performance level information for the data centers.

VmLoadBalancer. The responsibility of this component is to allocate the load on various data centers according to the request generated by users. One of the three given policies can be selected. The given policies are round robin algorithm, equally spread current execution load, and throttled.

CloudAppServiceBroker. The responsibility of this component is to model the service brokers that handle traffic routing between user bases and data centers. The service broker can use one of the routing policies from the given three policies which are closest data center, optimize response time and reconfigure dynamically with load. The closest data center routes the traffic to the closest data center in terms of network latency from the source user base. The reconfigure dynamically with load routing policy works in the sense that whenever the performance of particular data center degrades below a given threshold value then the load of that data center is equally distributed among other data centers.

4. Simulation Configuration

In order to analyze various load balancing policies configuration of the various component of the cloud analyst

tool need to be done. We have set the parameters for the user base configuration, application deployment configuration, and data center configuration as shown in figure 6, figure 7 and figure 8 respectively. As shown in figure the location of user bases has been defined in six different regions of the world. We have taken two data center to handle the request of these users. One data center is located in is located in region 0 and another in 2. On DC1 number of VM allocated are while that on DC2 are 50.

Simulation Duration: 180.0 min

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	8	1000	13	15	400000	40000
	UB2	1	12	1000	15	17	100000	10000
	UB3	2	8	1000	20	22	300000	30000
	UB4	3	9	1000	1	3	150000	15000
	UB5	4	7	1000	21	23	50000	100

Figure6. User Base Configuration

Application Deployment Configuration: Service Broker Policy: Closest Data Center

Data Center	#VMs	Image Size	Memory	BW
DC1	20	100	1024	10
DC2	50	100	1024	1000

Figure7 Application Deployment Configuration

Data Centers:	Name	Region	Arch	OS	VMM	Cost per VM \$/Hr	Memory Cost \$/s	Storage Cost \$/s	Data Transfer Cost \$/Gb	Physical HW Units
	DC1		0x86	Linux	Xen	0.1	0.05	0.1	0.1	2
	DC2		2x86	Linux	Xen	0.1	0.05	0.1	0.1	1

Figure8. Data center Configuration

5. Results

After performing the simulation the result computed by cloud analyst is as shown in the following figures. We have used the above defined configuration for each load balancing policy one by one and depending on that the result calculated for the metrics like response time, request processing time and cost in fulfilling the request has been shown.

5.1. Response Time:

Response time for each user base and overall response time calculated by the cloud analyst for each loading policy has been shown in the figure 9, 10 and 11 respectively. As can be seen from the figure the overall response time of Round Robin policy and ESCEL policy is almost same while that of

Throttled policy is very much low as compared to other two policies.

Overall Response Time Summary			
	Average (ms)	Minimum (ms)	Maximum (ms)
Overall Response Time:	754.81	67.97	1589.10
Data Center Processing Time:	472.77	0.40	1064.89

Response Time By Region			
Userbase	Avg (ms)	Min (ms)	Max (ms)
UB1	172.608	67.974	244.91
UB2	229.71	177.144	340.717
UB3	243.606	162.126	340.563
UB4	1,173.705	303.34	1,589.099
UB5	317.408	278.357	366.425
UB6	212.426	169.318	327.106

Figure9. Response time using Round Robin Policy.

Overall Response Time Summary			
	Average (ms)	Minimum (ms)	Maximum (ms)
Overall Response Time:	757.45	65.77	1580.88
Data Center Processing Time:	475.40	0.40	1053.09

Response Time By Region			
Userbase	Avg (ms)	Min (ms)	Max (ms)
UB1	172.897	65.767	244.378
UB2	229.507	177.144	356.591
UB3	243.925	162.126	350.241
UB4	1,178.218	303.34	1,580.882
UB5	318.247	278.357	375.242
UB6	212.526	169.318	329.052

Figure10. Response time using ESCEL Policy.

Overall Response Time Summary			
	Average (ms)	Minimum (ms)	Maximum (ms)
Overall Response Time:	511.33	63.30	1456.36
Data Center Processing Time:	245.06	0.40	935.23

Response Time By Region			
Userbase	Avg (ms)	Min (ms)	Max (ms)
UB1	117.943	63.3	194.085
UB2	225.715	177.144	328.153
UB3	160.77	72.256	303.224
UB4	781.09	299.542	1,456.362
UB5	318.209	278.357	374.415
UB6	210.389	169.318	336.314

Figure11. Response time using Throttled Policy.

5.2. Data Center Request Servicing Time:

Data Center Request Servicing Time for each data center calculated by the cloud analyst for each loading policy has been shown in the figure 12, 13 and 14 respectively. As can be seen from the figure the servicing time of Round Robin policy and ESCEL policy is almost same while that of Throttled policy is very much low as compared to other two policies.

Data Center Request Servicing Times			
Data Center	Avg (ms)	Min (ms)	Max (ms)
DC1	68.673	1.911	175.345
DC2	646.238	0.404	1,064.888

Figure12. Request Servicing time using Round Robin Policy.

Data Center Request Servicing Times			
Data Center	Avg (ms)	Min (ms)	Max (ms)
DC1	68.876	1.911	171.756
DC2	649.911	0.404	1,053.088

Figure13. Request Servicing time using ESCEL Policy.

Data Center Request Servicing Times			
Data Center	Avg (ms)	Min (ms)	Max (ms)
DC1	37.348	1.911	120.454
DC2	334.222	0.404	935.23

Figure14. Request Servicing time using Throttled Policy.

5.3. Processing Cost:

The processing cost for each load balancing policy computed by the cloud analyst as can be seen from the figures 15, 16 and 17 is same so, load balancing technique does not play any role in the processing cost.

Cost	
Total Virtual Machine Cost :	\$21.12
Total Data Transfer Cost :	\$483.08
Grand Total :	\$504.20

Figure15. Processing cost using Round Robin

Cost	
Total Virtual Machine Cost :	\$21.12
Total Data Transfer Cost :	\$483.08
Grand Total :	\$504.20

Figure16 Processing cost using ESCEL

Cost	
Total Virtual Machine Cost :	\$21.12
Total Data Transfer Cost :	\$483.08
Grand Total :	\$504.20

Figure17. Processing cost using Throttled

References

- [1] Anthony T.Velte, Toby J.Velte, Robert Elsenpeter, Cloud Computing A Practical Approach, TATA McGRAW-HILL Edition 2010.
- [2] A Survey on Open-source Cloud Computing Solutions Patrícia Takako Endo, Glauco Estácio Gonçalves, Judith Kelner
- [3] Cloud Analyst: A Cloud-Sim-based Tool for Modeling and Analysis of Large Scale Cloud Computing Environments. MEDC Project Report Bhatiya Wickremasinghe.
- [4] R. Buyya, R. Ranjan, and R. N. Calheiros, "Modeling and Simulation of Scalable Cloud Computing Environments and the CloudSim Toolkit: Challenges and Opportunities," Proc. of the 7th High Performance Computing and Simulation Conference (HPCS 09), IEEE Computer Society, June 2009.
- [5] <http://www.cloudbus.org/cloudsim>