International Journal of Computer Applications (0975 – 8887)
Volume 115 – No. 12, April 2015

An Enhancement in Service Broker Policy for Cloud-Analyst

Pushpi Rani
ITM University Gwalior

Reena Chauhan
ITM University Gwalior

Ritu Chauhan
ITM University Gwalior

ABSTRACT
Cloud computing means storing and accessing data and programs over the Internet instead of one's personal computer's hard drive. The cloud is an analogy for the Internet. Now a day, it becomes very popular due to new trend and technology. It deals with large amount of data, so, it is necessary to simulate the behavior of cloud in real time environment. For this purpose, the simulation tool such as, cloud-analyst, cloudSim, are commonly used, which has been provided by laboratory CLOUD. The simulator cloud analyst uses different load balancing policy, service broker strategy with different parameters and has used as per requirement. This paper presents an improvement in service broker strategy, which enhances the performance of data center.

Keywords
Cloudlet, simulation, CloudSim, processor utilization, efficiency

1. INTRODUCTION
Cloud Computing is a technology that experiencing rapid enhancement in world wide. It uses the internet and central remote servers to maintain data and applications. It allows consumers and businesses to use applications without installation and access their personal files at any computer with internet access. This technology allows for much more efficient computing by centralizing data storage, processing and bandwidth. Cloud computing comes into focus only when one thinks about what IT always needs: a way to increase capacity or add capabilities on the fly without investing in new infrastructure, training new personnel, or licensing new software. Cloud computing encompasses any subscription-based or pay-per-use service that, in real time over the Internet, extends IT's existing capabilities.

As the new technology is enhancing day by day, the cloud environment also becomes vast and it has to maintain the large volume of data. Here the main challenge is how to deal with these data in real time environment. For this purpose cloud simulator has been used. By using the simulation tool, it is very easy to imitate the real environment. It helps the application developers, and they are able to determine the strategy for allocation of resources among data center, this strategy is commonly known as load balancing [1] techniques. The other issue is to decide which data center will provide the service to the new request coming from user base and it is decided by the service broker strategy [1]. The aim of this paper is to design a new service broker strategy, which is more efficient than the existing algorithm, as well as it will improve the performance of data center.

2. CLOUD ANALYST
Cloud Analyst [2]. is an open source toolkit, having the feature of original framework and extending the capabilities of cloudSim [2].The main aim of CloudAnalyst is to separate the simulation experiment exercise from programming technicalities, so a modeler can focus on the simulation. It hides the programming complexity and provide a user friendly environment as a simulation toolkit. The CloudAnalyst [2,3] enables a modeler to repeatedly execute simulations and to conduct a series of simulation experiments with slight parameters variations in a quick and easy manner. These are the features of cloudAnalyst which make it popular in cloud environment:

- Easy to use Graphical User Interface (GUI)
- Ability to define a simulation with a high degree of configurability and flexibility.
- Repeatability of experiments.
- Graphical output.
- Use of consolidated technology and ease of extension

Figure 1: CloudAnalyst GUI

CloudAnalyst is built on top of CloudSim toolkit [3], by extending its functionalities with the introduction of concepts that model Internet and Internet Application behavior. CloudAnalyst components

GUI Package. It is responsible for the graphical user interface, and acts as the front end for the application, managing screen transitions and other UI activities.

Region In the CloudAnalyst „Regions‟ are there based on the 6 main continents in the World. It is useful to create a real environment for the large scaled testing in Cloud-Analyst [3].

Simulation. This component is responsible for holding the simulation parameters, creating and executing the simulation.

UserBase. A User Base models a group of users that is considered as a single unit in the simulation and its main responsibility is to generate traffic for the simulation. The modeler may choose to use a User Base to represent a single user, but ideally a User Base should be used to represent a larger number of users for the efficiency of simulation. [3]

DataCenterController. This component add new data center for cloudlets and controls the data center activities.

Internet. This component models the Internet and implements the traffic routing behavior.

InternetCharacteristics. This is used to define the characteristics of the Internet applied during the simulation, including the latencies and available bandwidths between regions, the current traffic levels,and current performance level information for the data centers.

VmLoadBalancer: VM Load balancer is useful to determine which VM should be assigned the requests (Cloudlet) [3] for processing. This component models the load balance policy used by data centers when serving allocation requests. Default load balancing policy uses a round robin algorithm, which allocates all incoming requests to the available virtual machines in round robin fashion without considering the current load on each virtual machine [4]. Additionally, it is also offered a throttled load balancing policy that limits the number of requests being processed in each virtual machine to a throttling threshold [5]. If requests are received causing this threshold to be exceeded in all available virtual machines, then the requests are queued until a virtual machine becomes available.

CloudAppServiceBroker. This component models the service brokers that handle traffic routing between user bases and data centers. The default routing policy routes traffic to the closest data center in terms of network latency from the source user base. In addition an experimental brokerage policy for peak load sharing is implemented on CloudAnalyst [6]. This routing policy attempts to share the load of a data center with other data centers when the original data center's performance degrades above a pre-defined threshold.

Another important component of CloudAnalyst is the network model [7]. Modeling of bandwidth is probably the most complex task, especially considering the nature of a network such as the Internet.

3. CLOUDAPPLICATION SERVICE BROKER

A service broker decides which data center should provide the service to the requests coming from each user base [5]. And thus, service broker controls the traffic routing between User Bases and Data Centers. Currently, Cloud-Analyst is with three types of service brokers each implementing a different routing policy [8,9].

3.1 Service Proximity based routing [5]

Here, the shortest path to the data center from the user base, depended on the network latency is selected and according to that, the service broker routes the traffic to the closest data center with the consideration of transmission latency.

3.2 Performance Optimized routing [5]

In this routing policy, service broker actively monitors the performance of all data centers, and based on that, directs traffic to the data center with best response time

3.3 Dynamically reconfiguring router [5]

This router has one more responsibility of scaling the application deployment depended on the current load it faces. This policy increases and decreases the no. of virtual machines allocated in the data centers. This will be done taking under consideration the current processing times and best processing time ever achieved.

Service Proximity based routing, also known as closest path, is effective only when the datacenters and user base are uniformly distributed. *Performance Optimized routing,* focus on optimal response time, but in this case the one data center get overloaded and another are idle. To overcome this problem, the proposed algorithm has been designed.

4. PROPOSED ALGORITHM

The computational sequence of step to choose the appropriate Datacenter is described below.

Calculate the Percentage of processor utilization, which is defined as the ratio of useful CPU time over total CPU time. It can be written as:P=(T/T+S), where, T=Useful CPU time and S=Idle CPU time. In this case the CPU, means VM so, the total utilization at any data center is calculated by $P_{DC}= P * \text{No. of VM}$.

Now consider the following matrices as a parameter to compute the efficiency of each datacenter individually.

Average response time per unit time, T_R

Average waiting time per unit time T_W

Workload (requests) to be serviced per second (Mbps) or a unit of time **W**

Throughput (Request / Sec), T_H

The average time of processing T_P

The number of requests executed per unit time N_E

The number of requests per unit time buffer N_B

The number of rejected requests per unit time N_R

Efficiency (η)

$=(((T_R+T_P-T_W)*P_{DC})T_H)/(N_E+N_B-N_R)$

Next define the threshold value of every data center. DataCenter-Controller maintains the efficiency list and threshold value of each datacenter .When DataCenter-Controller receives a new request, then the datacenter having highest efficiency would be chosen. Now, compare the threshold of datacenter with overall load i:e; coming workload plus previous load on that Datacenter. If the workload is less than the threshold value then the datacenter _ID returns to DataCenter-Controller, if it reaches the threshold value, then the request will be transferred to the closest data center. Again the same process would be repeated for this datacenter, the closest datacenter _ID returns to DataCenter-Controller only if the workload is less than the threshold value. Otherwise the workload will be store in the buffer queue. After each request has been processed, the DataCenter-Controller update its list. The designed algorithm is divided into two parts, algorithm1 is a preprocessing algorithm which calculates the efficiency of datacenter. The output generated by this algorithm is used as input in Algorithm 2. This algorithm returns the datacenter number to the cloudlets.

Algorithm 1 Preprocessing

Input DC configuration

Output Efficiency(η)

for *all DC*

Calculate the efficieny using equation,

P_{DC}*=P*No.of VM.*

$\eta = (((T_R+T_P-T_W) x P_{DC}) T_H)/(N_E+N_B-N_R)$

end for

return efficiency list

Algorithm 2 Data Center Selection

Input:Region number, Efficiency list, threshold

Output: destination DC number

for any new request, DataCenter-Controller do

DC←Min(efficiency_list)

if*(DC_load<threshold_value)*

then *return DC_ID ←DC*

else

DC←closest_DC_ID

if*((DC_load<threshold_value)*

then *Return closest_DC_ID*

else *store the request in Queue.*

end

5. SIMULATION AND RESULT ANALYSIS

Simulation has been done by using cloudAnalyst tool. In order to analyze different service broker policies the cloud analyst tool need to be set. As shown in figure 2 the location of user bases has been defined in six different regions of the world. Five user base and four data center has been created. One data center is located in region 0, second in region1and third in region 3 and fourth in region 2. Number of VM in DC1 is 25, in DC2 is 40, in DC3 is 50 and DC4 it is 5. The result of different service broker algorithm with these data set has been shown in table 1.

Other parameters are set as follows:

Simulation Duration: 60 Hours Application

User Grouping Factor in User Base: 1000

Request Grouping Factor: 100

Executable instruction length per request (bytes):500

Load balancing policy: Round Robin

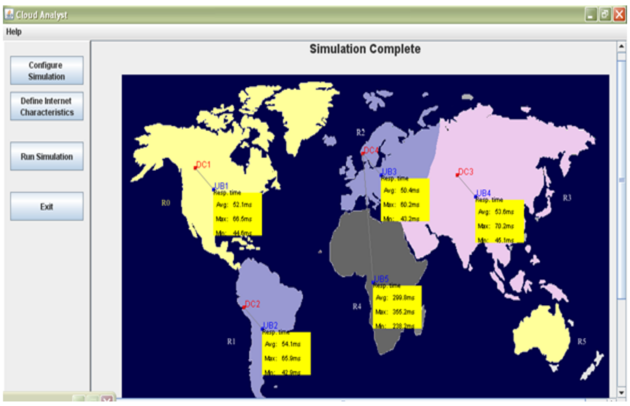

Figure 2: User base and Data center

Table 1: Data Center Request Servicing Time

VM in DC	Closest	Optimal	Proposed
25	2.09	1.95	2.17
40	4.09	4.7	4.9
50	3.08	3.02	3.19
5	1.05	1.05	2.05

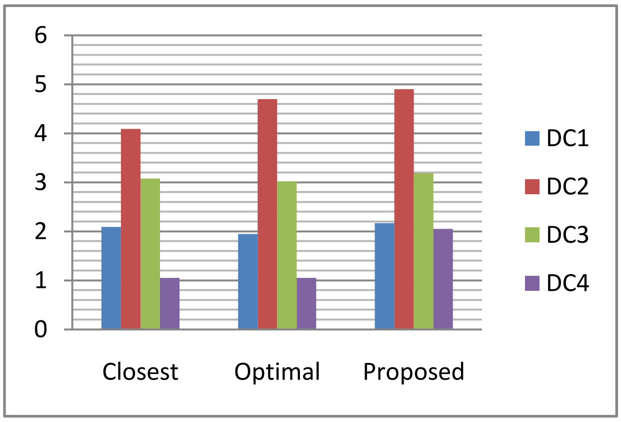

Figure 3: Comparison of proposed algorithm with existing algorithm.

6. CONCLUSION

In this paper, a new enhanced and efficient service broker policy is proposed and then implemented in cloud computing environment using CloudAnalyst toolkit. The proposed policy combines the features of closest path and optimal response time policy. By visualizing the graphs and tables it can easily identify that the data centre processing time is improved in comparison to the existing policies. The proposed algorithm does not include the changes occur in cloud. In future this work can be implemented in dynamic environment.

7. REFRENCES

[1] Tushar Desai, Jignesh Prajapati "A Survey Of Various Load Balancing Techniques And Challenges in Cloud Computing" International Journal Of Scientific & Technology Research, Volume 2, Issue 11, November 2013

[2] Bhathiya Wickremasinghe "Cloud Analyst: A Cloud-Sim-Based Tool For Modeling And Analysis Of Large Scale Cloud Computing Environments. MEDC Project", Report 2010.

[3] Bhathiya Wickremasinghe ,Roderigo N. Calherios "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, 2010.

[4] Zhang, L. Cheng, And R. Boutaba, "Cloud Computing: State-Of-The-Art And Research Challenges", Journal Of Internet Services And Applications, April 2010.

[5] Dhaval Limbani, Bhavesh Oza "A Proposed Service Broker Strategy in CloudAnalyst for Cost-Effective data center Selection"International Journal of Engineering

Research and Applications. Vol. 2, Issue 1, Jan-Feb 2012, pp.793-797

[6] 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.

[7] Jiyin Li, Meikang Qiu, Jain-Wei Niu,Yu Chen, Zhong Ming "Adaptive Resourse Allocation for Preemptable Jobs in Cloud Systems". IEEE International Conference on Intelligent Systems Design and Applications, pp 31-36, 2010

[8] R Mishra, S N Bhukya, "Service Broker Algorithm for Cloud-Analyst", International Journal of Computer Science and Information Technologies, Vol.5(3), 2014, 3957-3962

[9] Kishor Kunal, Thapar Vivek "An Efficient Service Broker Policy for Cloud Computing Environment", International Journal of Computer Science Trends and Technology. Vol. 2, Issue 4, July-Aug 2014.