Volume 5, Issue 6, June 2015 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

# Factor based Resource Allocation in Cloud Computing

**Satish Ranbhise**
M.E. (Computer Engg.)
Shri L.R.Tiwari College of Engg.
Mumbai, India

**Dr. J. W. Bakal**
Principal
S. S. JONDHALE College of Engg.,
Dombivli, India

***Abstract—Cloud computing is an area that is rapidly gaining popularity in both academia and industry. Cloud-Analyst is useful tool to model and analyze cloud computing environment and applications before actual deployment of cloud products. Service broker controls the traffic routing between user bases and data centers based on different service broker policies. Service proximity based routing policy selects closest data center to route the user request. If there are more than one data centers within the same region, it is selected randomly without considering workload, cost, processing time or other parameters. Randomly selected data center is prone to give unsatisfactory results in term of response time, resource utilization, cost or other parameters. In this paper we propose a service broker policy which distributes the requests based on the priority of data centers and gives better performance in cloud analyst.***

***Keywords : Cloud computing, Cloud-Analyst, Data Center, Service Broker Policy***

## I. INTRODUCTION

To meet cloud computing requirements, CloudSim was developed for modeling and simulation of extensible Clouds [3]. There are various software based algorithms are proposed to improve the performance of cloud. Various factors are used to determine and evaluate cloud performance such as: latency, throughput, CPU usage, network error, and response time. These factors highly depend on geographical distribution of data centers, number of users, and number of data centers, service broker policy being used and load balancing algorithms. Many changes have been done in the existing algorithms and many new algorithms are proposed to improve the performance.

For analyzing performance of cloud environment in various ways , actual deployment of cloud is costly and difficult task, therefore not preferable. For this reason various simulation tools are used to model and analyze cloud computing environment and applications, graphically analyze the results before the actual deployment of clouds. The simulation tools available are Cloud Reports, Cloud Analyst, MR-CloudSim and Network CloudSim etc. The underlying platform for these simulators is CloudSim[1].

## II. RELATED WORK

Different simulators have been proposed for modeling of Grid based environments [6]. Cloudsim is developed to simulate big Cloud environment for analyzing the cloud environment under various deployment configurations. Existing system simulators not applicable to the cloud environment. Analyzing the performance of cloud provisioning policies, services, application workload, models and resources performance models under varying system, user configurations and requirements is difficult to achieve. To tackle above problem, CloudSim can be used. Cloud-Analyst is an toolkit which helps us to simulate and assess the performance of cloud services. The simulator is developed using Java SE , Swing , CloudSim toolkit and SimJava framework[5].

The Various Service broker policies implemented in Cloud-Analyst are[8]:

1) Service Proximity Based Routing: In this routing policy service broker selects the shortest path from the user base to the data center, depending on the network latency and based on that, routes the traffic to the closest data center with the consideration of transmission latency.
2) Performance Optimized Routing: In this routing policy, service broker monitors the performance of all data centers, and based on that, routes the traffic to the data center with best response time.
3) Dynamically reconfiguring router: This is an extension to proximity based routing, where the routing logic is similar, but the service broker has one more responsibility of scaling the application deployment based on the load it is facing. This policy increases and decreases the number of virtual machines allocated in the data centers. Grouping of simulation elements help us to improve.

## III. NEED FOR EVALUATING CLOUD PERFORMANCE

From the routing of the user requests it is quite evitable that many of the issues arise while:

1. Selecting the appropriate data center: For Selecting the appropriate data center we have several service broker policies which affect the performance to a great extent. As it is first step towards providing better cloud performance. Therefore, properly selecting a data center by using appropriate Service broker algorithm is work of research.

 

2. Selecting appropriate VM: After selecting appropriate data center the Load balancing is the most important aspect in any data center. Load can be balanced only if proper VM allocation is done for a particular user request. Various Load balancing techniques are present and proposed to enhance the cloud performance.

The Situation may arise that all the requests may go to only one data center. As a result of that only one data center is highly loaded and others are not. Other situation may arise that there is a need of migrating VM or allocating new VM for a user request. There is a lot of research work going on for providing Service Broker policies and Load balancing algorithms which would increase/enhance Cloud Environment performance.

**CloudSim**

For analyzing the resource allocation in cloud computing environment which is scalable to n servers then we will require Cloud simulation and modeling tool which will create the cloud as per requirement[4] .

The primary objective of this project is to provide realistic, and extensible simulation framework that enables modeling, simulation, and experimentation of Cloud computing infrastructures and application services. By Choosing CloudSim, researchers developers can focus on system design issues that they want to investigate, without concerned about the details related to Cloud-based infrastructures and services. It also allows simulation of multiple Data Centers to study on distribution and associated policies for migration of VMs for reliability and automatic scaling of applications [2]. CloudSim toolkit is the tool for modeling large virtualized cloud environment which has components for data centers ,hosts, virtual machines, brokers and service requests[10].

**CloudReports**

CloudReports is a simulation tool that simulates distributed computing environments based on the Cloud Computing paradigm. The application simulates an Infrastructure as a Service (IaaS) provider with an different number of datacenters. Each datacenter is entirely customizable [9].

## IV. SERVICE PROXIMITY BASED ROUTING

It selects the Data Center with least network latency i.e. the closest DC. If two or more closest Data Centers are available then selection is done randomly. It does not take the load and cost into the consideration. The algorithm is as followed:

1. Service Proximity Service Broker maintains an index of Data Centers indexed by their region.
2. When the Internet gets a message from a concerned user base it queries the Service Proximity Service Broker for destination Data Center Controller.
3. The Service Proximity Service Broker retrieves the region of the sender of the request and queries for the region proximity list for that region from the Internet Characteristics.
   This list takes the remaining regions in the order of lowest network latency when calculated from given region.
4. The Service Proximity Service Broker picks the first data center located at the earliest/highest region in the proximity list. If more than one data center is located in a region, one is selected randomly.

The proposed algorithm works on the basis of service proximity service broker. Key aspects of algorithm are as follows:

**Phase 1 (Initialization):**

1. Assign the load to each datacenter using following formula: Factor of data Center=LCM of number of VMs in each Datacenter /No of VMs on that Datacenter
2. Create a regional Datacenter Index map with key as region and value as list of datacenters in that region.
3. Create a Map with key as region and values as pointer and load Counter that are initialized to zero. (Pointer variable runs through the DC List and Load Counter variable is counter for load of a datacenter)
4 Calculate the weight of Data Center and minus vm factor .

   Factor of a datacenter = $\frac{LCMofNumberofVM\ \sin eachdataCenter}{NumberofVMsonthatdatacenter}$

   Weight of datacenter =Factor of datacenter -α

   α =No of VMs

**Phase 2 (Routing):**

Figure 1 below shows the routing of user request in one of the Simulation tools i.e. Cloud-Analyst[7]

1. When the request arrives, the closest region is selected depending upon the proximity list .
2. Regional list (DcList) of datacenters is loaded from regional Data center Index map with corresponding closest region.
3. Pointer and load Counter values are loaded for corresponding closest region.
4. Select the data center in circular fashion keeping into consideration the proportion loads. For example if entry for number of virtual machines for DC1, DC2 and DC3 is 50, 30 and 10 respectively. So the corresponding proportion loads are in the ratio 3,5,15. Thus DC1 is assigned to process the first 3 cloudlets, DC2 is assigned to process the next 5 cloudlets and DC3 is assigned to process the next 15 cloudlets, out of first 23 cloudlets and the whole selection process is done in a repeated manner to process the entire set of workloads.

 

## V. ARCHITECTURE

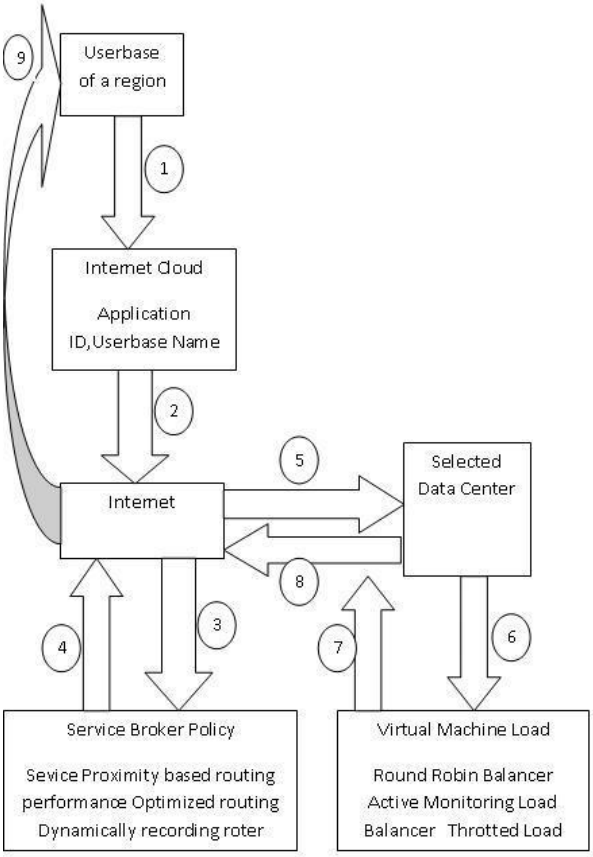


Figure 1: Existing System of cloud

(1)-Generate
(2)-Requests
(3)-Consults
(4)-Information about datacenter has been selected
(5)-Send request
(6)-Consults
(7)-Processing
(8)-Response
(9)-Towards appropriate userbase using "originator " field of Internet Cloudlet with associated delay.

## VI. RESULTS

TABLE 1. RESPONSE TIME BY DATACENTER

| Data Center | Avg (ms) | Min (ms) | Max (ms) |
|---|---|---|---|
| | | | |
| DC1 | 8.25 | 0.17 | 42.83 |
| | | | |
| DC2 | 35.04 | 0.58 | 63.50 |
| | | | |
| DC3 | 55.33 | 0.86 | 100.98 |
| | | | |
| DC4 | 57.05 | 0.88 | 100.99 |
| | | | |
| DC5 | 8.00 | 0.37 | 13.36 |
| | | | |
| DC6 | 38.44 | 2.57 | 63.35 |
| | | | |
| DC7 | 60.97 | 4.05 | 100.86 |

5. The pointer and load Counter values are updated for selected region.
6. Return datacenter name

 

TABLE 2. OVERALL RESPONSE TIME SUMMARY

**After applying Weight based allocation**

| | **Avg (ms)** | **Min (ms)** | **Max (ms)** |
|---|---|---|---|
| Overall response time: | 158.58 | 42.55 | 665.09 |
| Data Center processing time: | 20.43 | 0.17 | 100.99 |

TABLE3: RESPONSE TIME BY REGION

| **User base** | **Avg (ms)** | **Min (ms)** | **Max (ms)** |
|---|---|---|---|
| | | | |
| UB1 | 81.01 | 42.55 | 160.30 |
| UB2 | 217.35 | 161.07 | 333.15 |
| UB3 | 323.36 | 247.81 | 461.79 |
| UB4 | 69.56 | 46.22 | 158.64 |
| UB5 | 509.02 | 397.51 | 665.09 |
| UB6 | 213.69 | 163.24 | 328.13 |
| UB7 | 67.44 | 43.16 | 156.32 |
| UB8 | 67.58 | 44.11 | 157.31 |

TABLE 4 . : DATACENTER COST

| **Data Center** | **VM Cost $** | **Data Transfer Cost $** | **Total $** |
|---|---|---|---|
| DC6 | 150.85 | 2.01 | 152.86 |
| DC5 | 30.17 | 9.21 | 39.38 |
| DC4 | 288.02 | 3.82 | 291.84 |
| DC3 | 230.42 | 2.98 | 233.39 |
| DC2 | 120.01 | 3.73 | 123.74 |
| DC1 | 12.00 | 8.63 | 20.64 |
| DC7 | 241.36 | 1.33 | 242.69 |

The Results produced from the Cloud Analyst is shown in the following table:

TABLE 5. OVERALL RESPONSE TIME SUMMARY

| **Parameters** | **Avg (ms)** | **Min (ms)** | **Max (ms)** |
|---|---|---|---|
| Overall response time: | 177.09 | 43.16 | 696.59 |
| Data Center processing time: | 37.21 | 0.17 | 100.99 |

From the outputs, as shown in Table 2 , we can observe that proposed solution gives better overall response time than the conventional Random selection. We also observe that proposed algorithm has better data center processing time than the existing Service proximity based routing algorithm. Our proposed algorithm gives gives improvement in average data center processing time of 21.88ms improvement in average overall response time of 19.03ms against Data Center

 

processing time closest data center selection . The results of the execution show that the cost of the proposed algorithm has negligible difference with random selection done in Service Proximity based routing. We observe that the proposed solution yields a good amount of improvement in Overall average Response time and Data Center average processing time with the same cost. Thus the proposed algorithm leads to better resource utilization

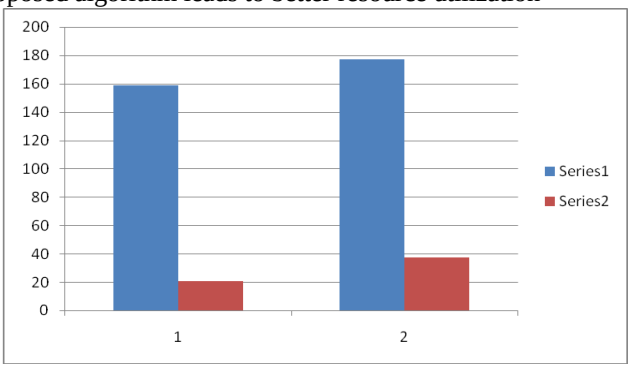


Figure2 : Graph of overall Response time and data center processing time

## VII. CONCLUSION

When it comes to efficient utilization of resource, it can be observed from the results that proposed algorithm utilizes resources well as compared to the existing random selection proximity based algorithm. The proposed algorithm gives improvement in average data center processing time of 21.88ms and improvement in average overall response time of 19.03ms against Data Center processing time closest data center selection . The results of the execution show that the cost of the proposed algorithm has negligible difference with random selection done in Service Proximity based routing. It is observed that the proposed solution yields a good amount of improvement in Overall average Response time and Data Center average processing time with the same cost. Thus the proposed algorithm leads to better resource utilization

The simulation results have shown that the proposed solution leads to a substantial reduction of response time in Cloud data centers in comparison to Proximity based service broker policy. We are aiming at meticulously analyzing the limitations of other service broker policies identified. In this project and come up with novel approaches to minimize or eliminate those limitations. Further work can be done to minimize the cost incurred. Certain other physical hardware characteristics can be combined to assign weights to the data centers. Also, Load balancing techniques can be applied for VMs in order to further optimize the utilization of the data centers in Cloud computing environments.

## REFERENCES

[1] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. F. De Rose, and R. Buyya, CloudSim: A Toolkit for Modeling and Simulation of Cloud Computing Environments and Evaluation of Resource Provisioning Algorithms, Software: Practice and Experience, 41(1): 2350, Wiley, January 2011

[2] Modeling and Simulation of Scalable Cloud Computing Environments and the CloudSim Toolkit: Challenges and Opportunities Rajkumar Buyya,Rajiv Ranjan and Rodrigo N. Calheiros

[3] CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms Rodrigo N. Calheiros, Rajiv Ranjan, Anton Beloglazov,C esar A. F. De Rose and Rajkumar Buyya

[4] CloudAnalyst: A CloudSim-based Tool for Modelling and Analysis of Large Scale Cloud Computing Environments Bhathiya Wickremasinghe

[5] RajuVadicharla, Ravi Kumar V, Nagaraju Samala, P.VenkatakishanRa A Open Source Tools Comparative Study on Cloud Computing" IN International Journal of Engineering Research and Development -ISSN: 2278-067X, p-ISSN: 2278-800X Volume 6, Issue 7 (April 2013), PP.69-73

[6] Priority Based Round-Robin Service Broker Algorithm for Cloud-Analyst Rakesh Kumar Mishra, Sandeep Kumar, SreenuNaik B IEEE International Advance Computing Conference (IACC) 2014.

[7] An Efficient Service Broker Algorithm for Data Center Selection in Cloud Computing Ku-nalKishor, VivekThaparin International Journal of Computer Science Trends and Technol-ogy (IJCST) Volume 2 Issue 4, July-Aug 2014

[8] R.Kanniga Devi , S.Sujan" A Survey on Application of Cloudsim Toolkit in Cloud Comput-ing " in InternationalJournal of Innovative Research in Science, Engineering and Technology Vol. 3, Issue 6, June 2014

[9] Colheiro R. N. CloudReport: A CloudSim - based Tool for Modelling and Analysis of Large Scale Cloud Computing Environments, MEDC Project by Rajkumar Buyya project supervisor Report

[10] Kavita Bhatt, Mahesh Bundele Department of Computer Science and Engineering Poornima College of Engg, Jaipur ,Rajasthan, India CloudSim Estimation of a Simple Particle Swarm Algorithm International Journal of Advanced Research in Computer Science and Software Engineering Volume 3, Issue 8, August 2013

 