ISSN:0975-9646

Rakesh Kumar Mishra et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (3) , 2014, 3957-3962

Service Broker Algorithm for Cloud-Analyst

Rakesh Kumar Mishra , Sreenu Naik Bhukya

Department of Computer Science & Engineering

National Institute of Technology Calicut, 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 this paper we propose priority and extended priority based Round- Robin service broker algorithms which distribute the requests based on the Rating of data centers and gives better performance than the conventional Random selection algorithm.**

***Keywords*—Cloud computing, Simulation, Cloud-Analyst, Service Broker Policy, Round-Robin, Rating Factor.**

I. INTRODUCTION

Cloud computing is the technology of future which provides “IT resources as a service” on demand of the user following the “pay-per-use”. The offered Cloud Service Models classified as Infrastructure as a service (IaaS), Platform as a service (PaaS), Software as a service (SaaS). Gmail, Google Docs, Microsoft Windows Azure, Google App Engine and Amazon Elastic Compute Cloud (EC2) are examples of cloud service. The offered Cloud Deployment models classified as public or private or combined of both [1]. Cloud computing is the internet based computing in which all computational operation is made to be performed over the cloud. We know that for resource management more cost need to be pay. So it is better to use the resources on rent basis rather than to buy our own resources. It is basically increasing the utilization of IT. Cloud based application cost is associated with two parameter virtual machine and data transfer corresponding to the user base [2]. These cloud architectures demand timely, repeatable, and controllable methodologies for evaluation of algorithms, applications, and policies before actual development of cloud services or products. It is very difficult to the researcher and industry user to identify the performance of internet based application using real cloud platform. Study of such distributed, virtualized, and dynamic resources can be carried out in a controlled manner with simulation [3].

For Cloud Computing environment, simulation-based approaches offer significant benefits, as it allows to test cloud services or products in repeatable and controllable environment free of cost, and to tune the performance bottlenecks before deploying on real Clouds. Simulation tool which are used for distributed application based on object oriented programming. Simjava, Gridsim, Cloudsim, Cloud-Analyst are the cloud simulation tool which provide the clue to us how to deploy application and what are the IT requirements for the application. These tools follow the layered architecture i.e. user can add their own layer over the user code level. Simulation tool provides the prior information about cloud resources which required for application deployment. We can use our own policy at data center level to share the MIPS of the physical processing element. Using simulation tool we can setup the different cloud configuration with internet characteristics. Processing power of the CPU to run their application is provisioned in time and space shared mode [4].

The remainder of this paper is organized as follows: In section II, we have discussed about the Cloud-Analyst. In section III, we have discussed about routing of user request in Cloud-Analyst. In section IV, we have discussed about working of service proximity based routing. Section V discusses about the related work done in this field. In section VI we propose algorithms to find the solution to the posed problem. Section VII discusses about the simulation configuration, results and performance analysis, and finally we have concluded our work.

II. CLOUD-ANALYST

Cloud-Analyst is an open source toolkit [5] which helps us to simulate and evaluate the performance of cloud services. Response time and data center processing time act as a performance evaluation parameter. Simulation results help us in improvement of quality of service. It is built on the top of Cloudsim [6]. In addition to the data center operations pro- vided by the Cloudsim toolkit, these are some new extensions in Cloud-Analyst [3].

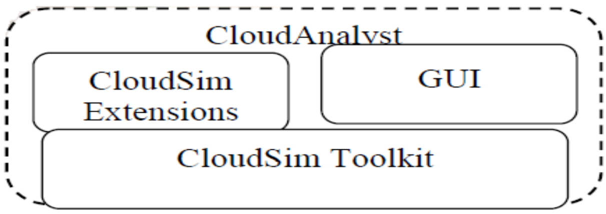

Fig. 1 Cloud-Analyst architecture

Application users: This component act as traffic generator. Its behavior is configurable

Internet: This component introduces to model data transmission across Internet with network delays and bandwidth restrictions.

Service Brokers: In Cloudsim DataCenterBroker performs VM management in multiple data centers and routing traffic to appropriate data centers. But in Cloud-Analyst these two main responsibilities were separated and assigned to DataCenterController and CloudAppServiceBroker.

GUI and ability to save simulations and results: Using GUI user can configure simulation with high level of details. It makes easy to do simulation experiments in repeatable manner. User can also save simulation configurations as well as results in the form of PDF files for future use.

A. Main Component

These are main components of Cloud-Analyst [3].

Region: The world is divided into 6 Regions based on the 6 main continents in the world. The other main entities such as user bases and data centers belong to one of these regions.

User Base: 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. A single user base may represent thousands of users but is configured as a single unit and the traffic generated in simultaneous bursts representative of the size of the user base. 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.

Internet: It is an abstraction for the real world Internet, implementing only the features that are important to the simulation. It 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.

Internet Cloudlet: It is a grouping of user requests. The number of requests grouped into a single Internet Cloudlet. It is having information such as the number of requests, the size of a request execution command, size of input and output files, the originator and target application id used for routing by the Internet.

Data Center Controller: This component controls data center activities such as VM creation and destruction and does routing of user requests received from user bases via the Internet to the VMs document is a template. An electronic copy can be downloaded from the conference website. For questions on paper guidelines, please contact the conference publications committee as indicated on the conference website. Information about final paper submission is available from the conference website.

VM Load Balancer : Data center controller uses VM load balancer to determine which VM should be assigned the next requests (Cloudlet) for processing. Three load balancing policies are currently included in the Cloud-Analyst.

1) *Round-Robin Load Balancer:* This load balancer uses a simple round-robin algorithm to allocate VMs.
2) *Active Monitoring Load Balancer:* This load balancer attempts to maintain equal workloads on all the available VMs.
3) *Throttled Load Balancer:* This load balancer ensures only a pre-defined number of Internet Cloudlets are allocated to a single VM at any given time. If more request groups are present than the number of available VM's at a data center, some of the requests will have to be queued until the next VM becomes available.

Cloud Application Service Broker: A service broker decides which data center should provide the service to the requests coming from each user base. And thus, service broker controls the traffic routing between user bases and data centers. Three service broker policies are currently included in the Cloud- Analyst.

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 actively monitors the perfor- mance 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 exten- sion 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 allo- cated in the data centers.

Grouping of simulation elements help us to improve the efficiency of simulation. In Cloud-Analyst the events are grouped at three levels. In the first level, there are user bases, which represent a cluster of users which are handled as a single unit. In the next level, user requests generated from each regional user base are grouped based on a grouping factor, which is kept independent of the user base size. In the last level, requests simultaneously processed by a single virtual machine are grouped. The last two grouping factors are configurable by Cloud-Analyst users, and it is also possible not to group simulation elements.

III. ROUTING OF USER REQUESTS

Routing of user requests in Cloud-Analyst is shown in the Fig. 2 including the use of service broker policy and the virtual machine load balancer [7].

1) User base generates an Internet Cloudlet, with appli-cation id for application and also includes name of the user base itself as *originator* for routing back the RESPONSE.
2) REQUEST is sent to the Internet with zero delay.
3) Internet consults the service broker for the data center selection. The service broker uses any one of the service broker policy based on the REQUEST.
4) Service broker sends information about selected data center controller to the Internet.
5) Internet adds appropriate network delay with the REQUEST and sends to the selected data center controller.
6) Selected data center controller uses any one of the virtual machines load balancing policy.

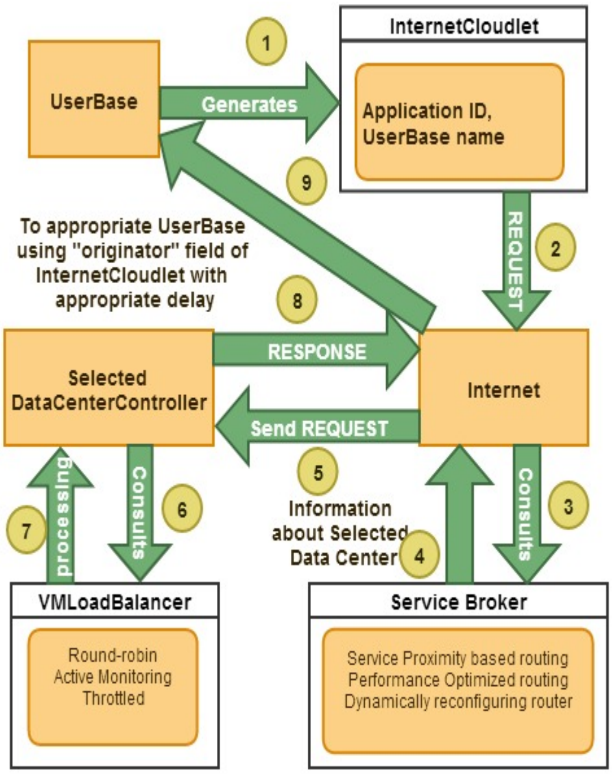

Fig. 2 Routing of User Requests

7) Virtual machines load balancer assign the virtual machine to the user request.
8) Selected data center sends the RESPONSE to the Internet after processing the REQUEST.
9) Internet uses the *originator* field of the Cloudlet in- formation and adds appropriate network delay with RESPONSE and sends to the user base.

IV. Working Of Service Proximity Routing

This is based on closest data center strategy [7]. Working of service proximity based routing is shown in Fig. 3.

1) ServiceProximityServiceBroker maintains an index table of all data centers indexed by their region.
2) When the Internet receives a message from a user base it queries the ServiceProximityServiceBroker for the destination DataCenterController.
3) The ServiceProximityServiceBroker retrieves the re- gion of the sender of the request and queries for the region proximity list for that region from the Internet Characteristics. This list orders the remaining regions in the order of lowest network latency first when calculated from the given region.
4) The ServiceProximityServiceBroker 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.*

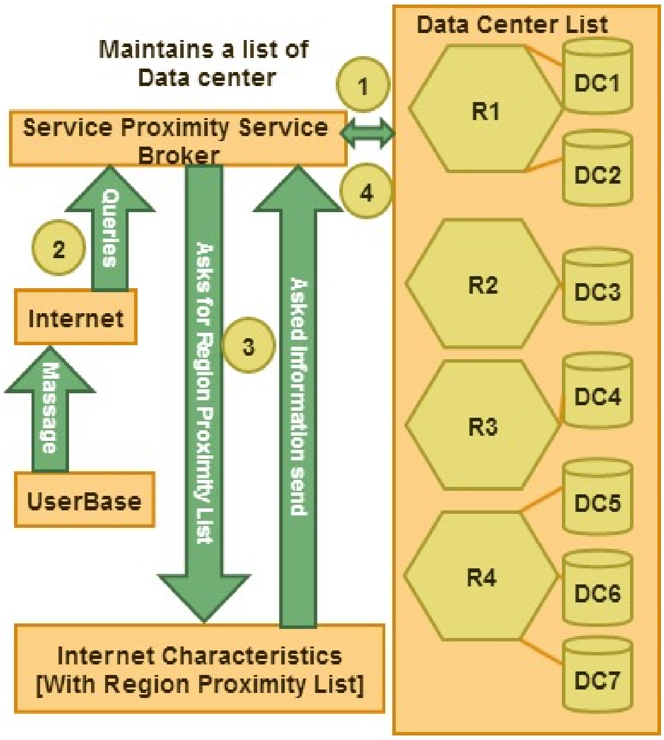

Fig. 3 Service Proximity based Routing

A. *Problems In Service Proximity based Routing*

1) When there are more than one data centers in the same region Random selection of data center.
2) For the same configuration, results may be different and developers/researchers may get difficulties to use the results.
3) There is possibility that selected data center will have higher workload, processing time, response time or cost etc.
4) There is also possibility of under-utilization of the resources.

V. Related Work

There are many problems in service proximity based routing algorithm due to random selection of data center. In order to overcome these problems instead of selecting data center randomly, we can select data center which is better in terms of processing time, response time, workload, cost etc. In 2012, Dhaval et al .[7] proposed an extension which select the most cost-effective data center. This reduces the overall cost but increases the response time. In 2012, A. Sarfaraz [8] proposed an extension which selects data center with high configuration. This reduces the response time but increases the overall cost. Both of these solutions results under-utilization of resources, Because each time service broker selects same data center (the lowest costly in first case and the highest costly in second case) so other data center will never be selected.

VI.. Proposed Solution

Service proximity based routing algorithm selects data center randomly when there are more than one data center in the closest region. So there are many problems due to random selection of data center within the region. In order to overcome these problems instead of selecting data center randomly, we can select data center in Round-Robin manner to distribute requests uniformly among all the data centers within a region. It leads to more resource utilization, but data centers may have different processing speed and cost. There are some data centers which are faster than others so these data centers should have been selected more number of times than slower ones to get better performance and resource utilization. If we want to reduce the overall cost then we have to select less costly data center more number of times than others. So we have to select the data center in proportion of their Rating.

We have used Preprocessing algorithm in the beginning of the simulation to create priority list. This list contains sequence of index of data center indexed by their region. Preprocessing step will be executed only once. After Preprocessing, Data Center Selection algorithm 1 uses this priority list to select the data center for each region.

.

Algorithm 1 Data Center Selection

```
Input: Region number, prioritylist
Output: Destination DC name
Steps:
Dclist ← regionalDataCenterIndex.get(region)
if Dclist is not NULL then
    noDc ← Dclist.size()
    if noDC==1 then
        DcName ← Dclist.get(0)
    else
        index← prioritylist.getnextindex(region)
        DcName ← Dclist.get(index)
    end if
end if
return DcName
```

A. Priority based Round-Robin Selection

In this we select the data center in proportion of their Rating. By using *eq^n 1* we can calculate the Rating of all n data center located at any region r.

$$R_{ri} = \frac{SPD_{ri}}{\min\left(SPD_{r1}, SPD_{r2}, ..SPD_{rn}\right)} \qquad (1)$$

In *eq^n 1*, R_{ri} is the Rating and SPD_{ri} is the speed per dollar of i^{th} data center located at region r. SPD is the ratio of processor's speed to cost of that data center. It leads to more resource utilization and performance. If further improvement is required, such as there is an absolute need to reduce the overall all cost of the deployment, at the cost of acceptable decrease in performance, then we will have to use *eq^n 2* with varying values of ***R*** (Rating factor) for rating calculation (extended priority based Round-Robin selection). Yet, this reduced performance is found to be still better than what is achieved by the conventional algorithm.

Preprocessing Algorithm 2 is used for preprocessing. After preprocessing, Data center selection algorithm 1 is used for selecting the data center for each region.

Algorithm 2 Preprocessing

```
Input: DC configuration
Output: prioritylist
for all region having more than one data center do
    Calculate the Rating of each data centers using eq^n 1.
    Generates a sequence of index for which the data center
    gets used as many times as the Rating assigned to
    it(follow highest Rating first order).
    Store these sequence indexed by their region in prior-
    itylist.
end for
return prioritylist
```

B. Extended Priority based Round-Robin Selection

In this we select the data center in proportion of their Rating. By using *eq^n 2* we can calculate the Rating of all n data center located at any region r.

$$R_{ri} = \frac{S_{ri} * R}{min(S_{r1}, .S_{rn})} + \frac{\max(C_{r1}, .C_{rn}) * (1 - R)}{C_{ri}} \qquad (2)$$

In *eq^n 2*, S_{ri} is the speed & C_{ri} is cost of i^{th} data center located at region r. ***R*** is the rating factor and its value varies from 0.0 to 1.0. By changing the value of ***R*** we can change the Rating of data center and hence the behaviour of our algorithm. It provides different levels of service and helps us to improve the quality of service (QoS). We have now modified the Simulator in which we can set the value of ***R***. The optimal value of ***R*** cannot be pre-determined. To find the optimal value of ***R***, we have to run the simulator with varying values of ***R*** (0.0-1.0) and then observe the results.

Preprocessing Algorithm 3 with an extra parameter ***R*** (Rating factor) is used for preprocessing. After preprocessing, Data center selection algorithm 1 is used for selecting the data center for each region.

Algorithm 3 Preprocessing

```
Input: DC configuration, Rating Factor (R)
Output: prioritylist
for all region having more than one data center do
    Calculate the Rating of each data centers. using eq^n 2.
    Generates a sequence of index for which the data center
    gets used as many times as the Rating assigned to it
    (follow highest Rating first order).
    Store these sequence indexed by their region in prior-
    itylist.
end for
return prioritylist
```

VII. Simulation And Results

We have used the Cloud-Analyst tool to evaluate the service proximity based routing algorithm, with Random, priority based and extended priority based Round-Robin data center selection one by one.

A. Simulation Configuration

1) User Base Configuration: The design model uses the user base to represent the single user but ideally a user base should be used to represent a large numbers of users for efficiency of simulation. Table I shows the user base configuration.

Table I. USER BASE CONFIGURATION

UB name	R	Peak Hrs.(GMT)	Avg. users	
			Peak	Off peak
UB1	0	3:00-9:00	10000	5000

2) Data Center Configuration: Data center manages the data management activities and does the routing of user requests received from user base via the internet to virtual machines. Table II shows data center configuration

Table II. DATA CENTER CONFIGURATION

Name	R	Cost/Vm ($/Hr)	Data transfer Cost ($/GB)	Speed (MIPS)
DC1	0	0.08	0.08	500
DC2	0	0.09	0.09	700
DC3	0	0.10	0.10	1000
DC4	0	0.11	0.11	1300
DC5	0	0.12	0.12	1600

4) *Other Configuration*: Table III shows the other parameters used in simulation

.

Table III. OTHER PARAMETERS

Parameter	Value
User Grouping Factor in User Base	1000
Request Grouping Factor	10
Executable instruction length/request	500
Load Balancing Policy	Throttled
Simulation Duration	24Hrs.
VM Image Size	10000
VM Memory	512Mb
VM Bandwidth	1000
Data Center Architecture	X86
Data Center processor/machine	4
Data Center OS	Linux
Data Center VMM	Xen

B. Priority based Round-Robin Selection

1) Result: The different cases have been prepared using the configurations in the Table II. For each case, we can select any 4 of the data centers from it, which may not necessity be distinct. Now, each Random and priority based Round- Robin selection algorithm has been run with these various configurations. The simulation result is shown in Table IV. ART is average response time of all user base and APT is average processing time of all data center. For each case *1st* row (Rand.) represents result of Random selection and *2nd* row (PBRR) represents result of priority based Round-Robin selection.

Table IV. OUTPUT RESULT

Case	Select.	ART(ms)	APT(ms)	Cost($)
4XDC3	Rand.	149.97	100.2	133.17
	PBRR	112.07	62.30	133.17
DC1,3XDC5	Rand	142.17	92.16	136.42
	PBRR	95.95	46.16	139.50
2XDC1,2XDC5	Rand.	182.75	113.2	123.26
	PBRR	105.46	55.67	129.20
3XDC1,DC5	Rand.	214.68	165.0	111.14
	PBRR	123.68	75.52	116.39
2XDC3,2XDC5	Rand.	129.85	80.61	136.14
	PBRR	99.16	49.38	135.93
2XDC3,DC4,DC5	Rand.	134.00	84.23	132.86
	PBRR	101.38	51.56	132.83
DC1,DC2,DC3,DC4	Rand.	177.70	128.1	117.26
	PBRR	119.34	69.65	120.20

2) Performance Analysis: The response time and the data center processing time are the two parameters to measure the performance.

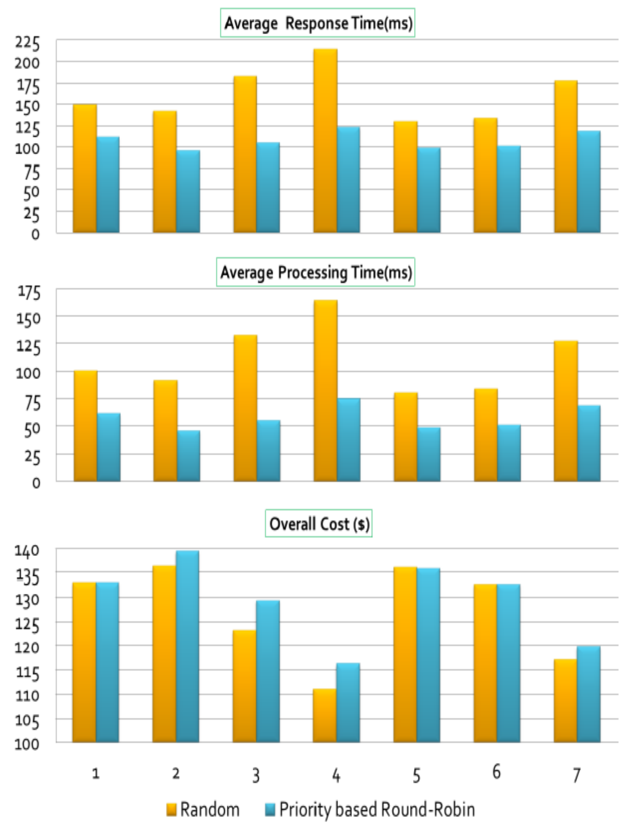

Fig. 4 Performance Analysis

From the Fig. 4, we can observe that proposed algorithm gives better response time than the conventional Random selection in all cases. We can also observe that the proposed algorithm gives better data center processing time than the conventional Random selection algorithm in all cases. We also observed that in some cases the proposed priority based Round-Robin selection algorithm costs slightly more than the conventional Random selection algorithm, the small increase in cost enables us to get far more improved results in processing time and response time. But, even this extra cost can be reduced if we use eq^n *2* in the Preprocessing Step (Extended priority based Round-Robin selection algorithm).

Resource utilization of Random selection algorithm depends on distribution of random number. Proposed priority based Round-Robin selection algorithm is distributing requests based on rating of data centers within a region. It leads to more resource utilization.

C. Extended Priority based Round-Robin Selection

1) Result: In this case, 5 data center with different configuration have been used for the purpose of simulation. Now, three different algorithms have been run with this configuration. Also, the extended priority based Round- Robin selection algorithm has been run for varying values of ***R*** (0.0-1.0). Table V shows the result of simulation.

Table V.OUTPUT RESULT

DC Selection	**DC Rating**		**ART (ms)**	**APT (ms)**	**Cost($)**
Rand.			152.41	102.6	133.35
PBRR	***R***	1,1,1,2,2	92.55	42.80	139.51
Ex-PBRR	0.0	2,2,1,1,1	148.73	98.96	125.88
	0.1	2,2,2,1,1	131.19	82.43	126.70
	0.2	2,2,2,2,2	96.54	46.68	133.21
	0.3	2,2,2,2,2	96.54	46.68	133.21
	0.4	2,2,2,2,2	96.54	46.68	133.21
	0.5	2,2,2,2,3	92.40	42.53	136.25
	0.6	2,2,2,3,3	92.42	42.63	137.38
	0.7	1,2,2,3,3	92.66	42.86	140.97
	0.8	1,2,2,3,4	95.16	45.33	143.10
	0.9	1,2,2,3,4	95.16	45.33	143.10
	1.0	1,2,3,4,5	98.18	48.40	144.52

2) Performance Analysis: From the output, as shown in Table V, we can observe that as the value of ***R*** is changed, the Rating of the data center gets changed and hence the behavior of our algorithm also changed, as a result of which the response time, data processing time and cost change.

From the Fig. 5, we can observe that for ***R*** (0.0-0.4) extended priority based Round-Robin selection algorithm gives better response time and processing time at lower cost and for ***R*** = 0.4 extended priority based Round-Robin selection algorithm gives best performance at lower cost when compared with Random selection algorithm. For ***R*** (0.5-0.6) extended priority based Round-Robin selection algorithm gives almost same performance as priority based Round-Robin selection algorithm at lower cost. We can find the optimal value of ***R*** for the desired performance and cost. It depends on the configuration and pricing plan of data center. We can easily obtain better values of performance at lower cost using this algorithm.

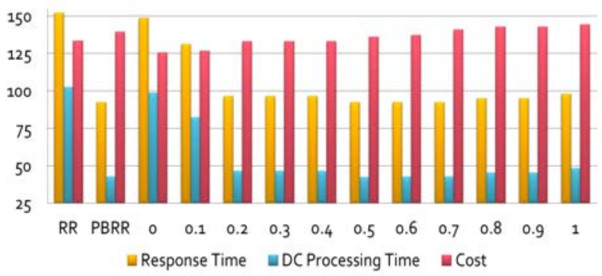

Fig. 5 Performance Analysis

VIII. CONCLUSIONS

There were many problems in service proximity based routing algorithm due to random selection of data center. So we have proposed priority based and extended priority based Round-Robin selection algorithm. From the results of simulation it can be concluded that proposed algorithm works efficiently when it comes to resource utilization, processing time of the data center and response time of user base. In proposed extended priority based Round-Robin selection algorithm we can find the optimal value of ***R*** (rating factor), which enables us to get desired performance and cost. These algorithms can be improved in future by introducing new equation for the rating calculation.

REFERENCES

[1] Christopher S Yoo. Cloud computing: Architectural and policy implications. Review of Industrial Organization, 38(4):405–421, 2011.

[2] Bhathiya Wickremasinghe, Rodrigo N Calheiros, and Rajkumar Buyya. Cloudanalyst: A cloudsim-based visual modeller for analysing cloud computing environments and applications. In Advanced Information Networking and Applications (AINA), 2010 24th IEEE International Conference on, pages 446–452. IEEE, 2010.

[3] Bhathiya Wickremasinghe and Rajkumar Buyya. Cloud- analyst: A cloudsim-based tool for modelling and analysis of large scale cloud computing environments. MEDC Project Report, 2009.

[4] Pradeep Singh Rawat, GP Saroha, and Varun Barthwal. Performance evaluation of social networking application with different load balancing policy across virtual machine in a single data center using cloudanalyst. In Parallel Distributed and Grid Computing (PDGC), 2012 2nd IEEE International Conference on, pages 469–473.

[5] Cloud-analyst can be downloaded from here http://www.cloudbus.org/cloudsim.

[6] Rodrigo N Calheiros, Rajiv Ranjan, César AF De Rose, and Rajkumar Buyya. Cloudsim: A novel framework for modeling and simulation of cloud computing infrastruc- tures and services.

[7] Dhaval Limbani and Bhavesh Oza. A Proposed Service Broker Strategy in Cloudanalyst for Cost-effective Data Center Selection. In International Journal of Engineering Research and Applications, India, Vol. 2, Issue 1, pages 793–797, Feb 2012.

[8] A. Sarfaraz Ahmed. Enhanced Proximity-based Routing Policy for Service Brokering in Cloud Computing. In International Journal of Engineering Research and Ap- plications, India, Vol. 2, Issue 2, pages 1453–1455, Apr 2012..

[9] Manoranjan Dash, Amitav Mahapatra, and Narayan Ran- jan Chakraborty. Cost Effective Selection of Data Center in Cloud Environment. In International Journal on Advanced Computer Theory and Engineering (IJACTE), pages 2319–2526, 2013.

[10] Wei Zhao, Yong Peng, Feng Xie, and Zhonghua Dai. Modeling and simulation of cloud computing: A review. In Cloud Computing Congress (APCloudCC), 2012 IEEE Asia Pacific, pages 20–24.