Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 3, Issue. 1, January 2014, pg.355 – 365

RESEARCH ARTICLE

Efficient Service Broker Algorithm for Data Center Selection in Cloud Computing

Prof. Deepak Kapgate
Department of CSE, G.H.R.A.E.T
Nagpur (M.S.), India
deepakkapgate32@gmail.com

***Abstract* - In cloud computing, load balancing is required to distribute the local workload evenly across all the nodes. It helps to achieve a high user satisfaction and resource utilization ratio by ensuring an efficient and fair allocation of every computing resource. Proper load balancing aids in minimizing resource consumption, implementing fail-over, enabling scalability, avoiding bottlenecks etc. In this paper, we proposed and implemented new service broker (DC selection) algorithm in cloud computing. Also we compare the results of proposed technique with existing technique. This study concludes that the proposed DC selection algorithm mainly focus on reducing associated overhead, service response time and improving performance etc. Various parameters are also identified, and these are used to compare the existing techniques.**

***Keywords* - Cloud Computing; Load Balancing Techniques; Data Center Selection Algorithm; Service Broker Policy**

I. INTRODUCTION

In recent years, online applications have started becoming abundant with applications in various categories such as entertainment, health, games, business, social networking, travel and news. In recent years, this problem has been addressed by researchers though cloud computing. Cloud computing can be defined as the aggregation of computing as a utility and software as a service [1]. Where the applications are delivered as services over the Internet and the hardware and systems software in data centers provide those services [2]. Also called 'on demand computing', 'utility computing' or 'pay as you go computing', the concept behind cloud computing is to offload computation to remote resource providers. The key strengths of cloud computing can be described in terms of the services offered by cloud service providers: software as a service (SaaS), platform as a service (PaaS), and infrastructure as a service (IaaS) [3].

© 2014, IJCSMC All Rights Reserved

Cloud computing has widely been adopted by the industry, though there are many existing issues like Load Balancing, Virtual Machine Migration, Server Consolidation, Energy Management, security, etc. that are not fully addressed [4]. Central to these issues is the issue of load balancing that is a mechanism to distribute the workload evenly to all the nodes in the whole cloud to achieve a high user satisfaction and resource utilization ratio. The present problem with cloud computing is that bottlenecks of the system which may occur due to load imbalance, computing resource distribution inefficiently, Minimum resource consumption. So, Proper load balancing techniques not only helps in reducing costs but also making enterprises as per user satisfaction [5] [6]. Scalability, one of the very important features of cloud computing, is also affected by load balancing. Hence, improving resource utility and the performance of a distributed system in such a way will reduce the energy consumption require efficient load balancing.

In cloud computing load balancing aspect is divided into two broad categories as Data Center Selection so called **DataAppServiceBroker** and Virtual Machine Management (VMM) at each data center so called **DataCenterController**. The paper mainly focuses on implementation of Efficient Service Broker algorithm in which the effective selection of data center for upcoming request is done based on their processing capability. This research shows that how the effective service broker algorithm leads to minimization of load on data centers and reduction in response time felt by users.

The rest of the paper is organized as follows: Section II discusses about the existing load balancing (Service Broker) algorithms. Section IV, explains proposed algorithm and flow chart of proposed algorithm. Section 5 shows the implementation details of algorithm and says about working environment. Section 6 shows the results and comparison analysis. Finally Section 7 concludes the paper with future scope.

II. Literature Survey

Load Balancing Algorithms in cloud computing environment generally divide in two categories [13] as Static Load Balancing Algorithms and Dynamic Load Balancing Algorithm [14].

A. Static Load Balancing Algorithm

Static Load balancing algorithms assign the tasks to the nodes based only on the ability of the node to process new requests but they do not consider dynamic changes of these attributes at run-time, in addition, these algorithms cannot adapt to load changes during run-time. The process is based solely on prior knowledge of node’s processing power, memory and storage capacity, and most recent known communication performance.

Round Robin (RR) and Weighted Round Robin (WRR) are most commonly Static Load Balancing Algorithm used in Cloud Computing. Round Robin Algorithm does not consider server availability, server load, the distance between clients and servers and other factors. In this algorithm server selection for upcoming request is done in sequential fashion. The main problem with this approach is inconsistent server performance which is overcome by WRR. In WRR the weights are added to servers and according to weight amount of traffic directed to servers however for long time connections it causes load tilt.

B. Dynamic Load Balancing Algorithm

Dynamic Load Balancing Algorithms considers a combination of knowledge based on prior gathered information about the nodes in the Cloud and run-time properties collected as the selected nodes process the task’s components. These algorithms assign the tasks and may dynamically reassign them to the nodes based on

© 2014, IJCSMC All Rights Reserved

the attributes gathered and calculated. However, they are more accurate and could result in more efficient load balancing than Static Load Balancing Algorithm.

Least Connection (LC) and Weighted Least Connection (WLC) are commonly used dynamic load balancing algorithm. In LC the total no of connections on server are identified at run time and the incoming request is sent to server with least number of connections. However LC does not consider service capability, the distance between clients and servers and other factors. WLC considers both weight assigned to service node $W(S_i)$ and current number of connection of service node $C(S_i)$ [15][16]. The problem with WLC is as time progresses static weight cannot be corrected and the node is bound to deviate from the actual load condition, resulting in load imbalance.

Xiaona Ren et. al. [17] proposed prediction based algorithm called as Exponential Smoothing forecast-Based on Weighted Least-Connection (ESBWLC) which can handle long-connectivity applications well. In this algorithm the load on server is calculated from parameters like CPU utilization, memory usage, no of connections, size of disk occupation. Then load per processor (Load/p) is calculated and this algorithm uses (Load/p) as historical training set, establishes prediction model and predicts the value of next moment. The limitation with this algorithm is this algorithm does not consider the distance between client and servers, network delay and other factors. Deepak Kapgate et. Al. [25] proposed Extended**- ESBWLC** which overcomes above limitation. In this algorithm author directly calculate the response time at client side. This got response time is store for further reference. The response time at time instance 't+1' is predicted by using current response time at time instance 't' and previously predicted response time for time instance 't'.

In this paper the author is proposing static service broker algorithm which gives improved result in terms of reduction in data center loading, reduction of data centre request timing and reduction of costing of VM and data transfer. Here the author improves the service broker algorithm called service proximity service broker.

III. Proposed Service Broker Algorithm

The proposed service broker algorithm is created by combining the advantages of Service Proximity Service Broker and Weighted Round Robin Service Broker Algorithm. Service Proximity Service Broker is the simplest Service Broker implementation. In Service Proximity Service Broker the **earliest region** is selected based on the minimum communication delay and maximum available bandwidth from user base (client) to data center residing region. The region selection is based on the earliest/ highest region in the proximity list and any data center of the selected region is then selected randomly for the user requests to be processed [4] [6]. In Weighted Round Robin Service Broker algorithm the data centre selection is done based on the processing capacity of respective data center.

© 2014, IJCSMC All Rights Reserved

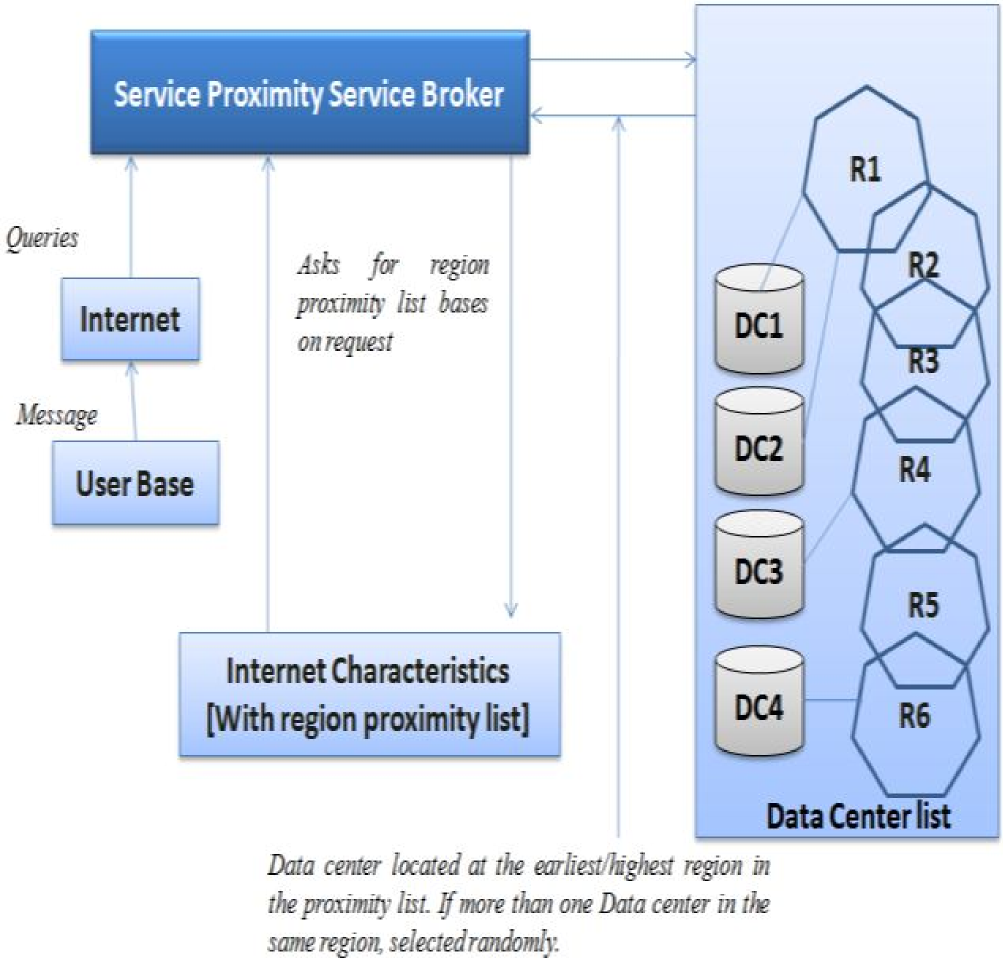

Figure 1. Service Proximity Based Routing

Problems using Service Proximity

1) Selection of data center is done randomly when more than one data center in the same region.
2) There is a possibility of selection of data center with higher cost.
3) For the same configuration, results may be different (random selection) and developers/researchers may get difficulties to use the results.

© 2014, IJCSMC All Rights Reserved

The flowchart for Proposed Prediction Algorithm:

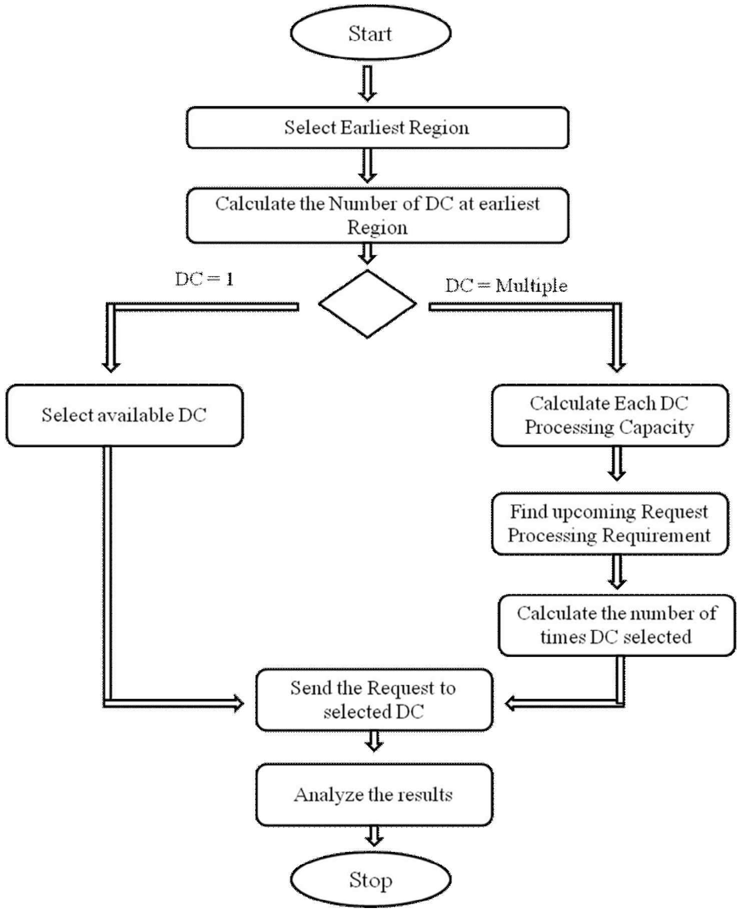

Figure 2. Flowchart of Proposed Algorithm

Proposed Algorithm:

1) Calculate the DC region with minimum communication delay and maximum usable bandwidth between user base (client) and data center as **Earliest Region**.

2) Find total number of data centers available at Earliest Region to satisfy upcoming request.

3) If there is single available data centre at Earliest Region then select it to satisfy the upcoming request.

4) If there is multiple data centers at Earliest Region then calculate the Processing Capacity of each data center.

5) Select the number of times the Data Centre is to be selected based on DC processing capacity and instruction and memory requirement of upcoming request.

6) Analyze the results.

© 2014, IJCSMC All Rights Reserved

V. Implementation Details

The working environment for cloud computing where the proposed algorithm is implemented is done using cloud analyst simulator which is built above “CloudSim”, “GridSim” and “SimJava”. Cloud-Analyst is built on the top of Cloud-sim. Cloud-sim is developed on the top of the Grid-sim.

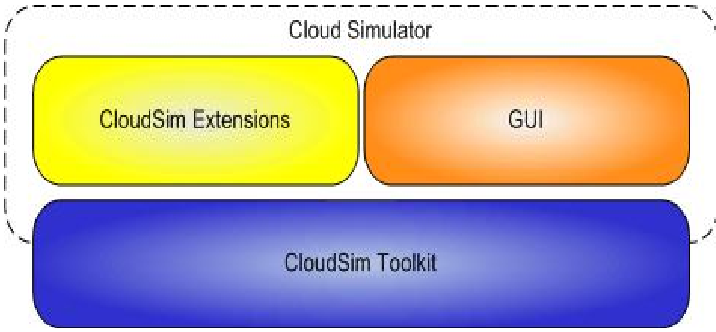

Figure 3. Cloud-Analyst built on top of Cloud-Sim toolkit

- Application users - There is the requirement of autonomous entities to act as traffic generators and behavior needs to be configurable.
- Internet - It is introduced to model the realistically data transmission across Internet with network delays and bandwidth restrictions.
- Simulation defined by time period - In Cloud-sim, the process takes place based on the pre-defined events. Here, in Cloud-Analyst, there is a need to generate events until the set time-period expires.
- Service Brokers - DataCeneterBroker in CloudSim performs VM management in multiple data centers and routing traffic to appropriate data centers. These two main responsibilities were segregated and assigned to DataCenterController and CloudAppServiceBroker in Cloud-Analyst.

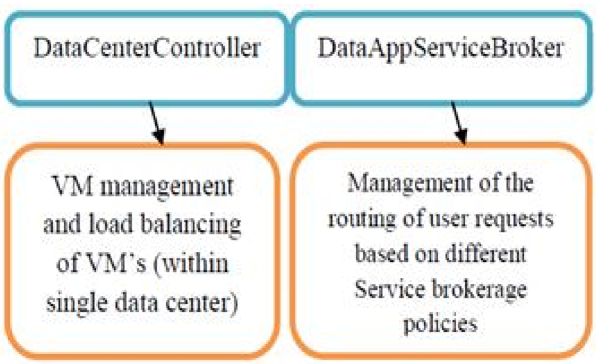

Figure 4. Responsibilities- Segregation

Routing of User Requests

In Cloud-Analyst, how the routing of user request takes place is shown in the figure (figure 2) below including the use of service broker policy and the virtual machine load balancer. [26][27]

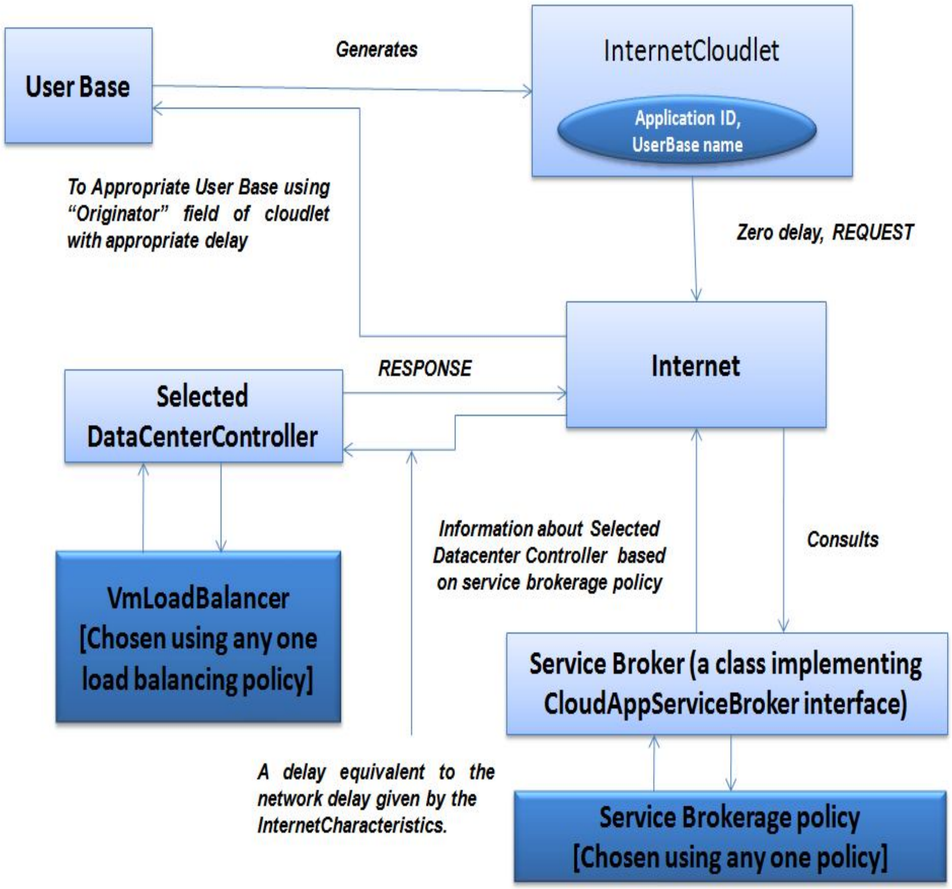

Figure 5. User Requests Routing

User Base generates an Internet Cloudlet, with the application id for the application it is intended and also includes the name of the User Base itself as the originator for routing back the responses. With the Zero delay, REQUEST is sent to the Internet. On receiving the REQUEST, 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 information and sends information about selected data center controller to the Internet. Using this information, Internet sends the REQUEST to the Data Center Controller. Now Selected Data Center Controller uses virtual machines load balancer and after processing the requests, sends the RESPONSE to the Internet. Now Internet will use the "originator" field of the cloudlet information it received earlier and will add appropriate network delay with RESPONSE and sends to the User Base.

VI. Results Calculated

The Proposed algorithm is implemented using simulation Cloud-Analyst. The scenario is taken where the data centers are located at single regions with user bases requesting services from different regions. The simulation runs approximately 60 min amount of time and the final result screen shown below as -

© 2014, IJCSMC All Rights Reserved

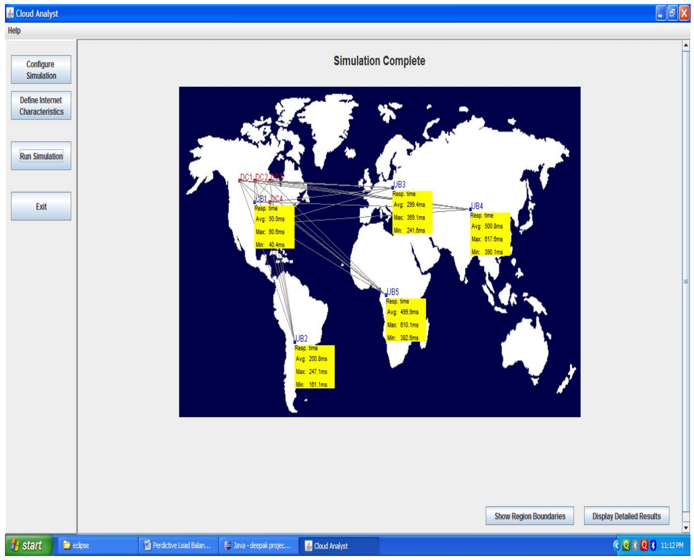

Figure 5. Cloud Analyst Main Result Screen

In above screen the lines shows that the user base is requesting service from corresponding data center or server. The values shown at boxes at each user bases represents the response time observed by respected user base. The values are the minimum response time calculated at client side wile requesting service from data center in the duration of simulation was running ,similarly it shows the maximum response time and the average response time from above two calculated values.

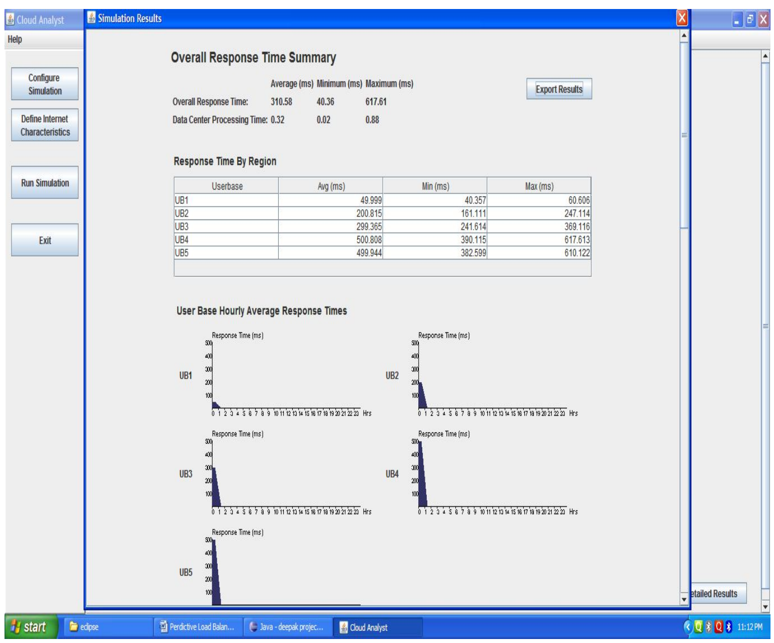

Figure 6. Result Screen Showing UB Response Time

The Simulation also shows the user base hourly average response time plotted on graph as shown in above figure.

© 2014, IJCSMC All Rights Reserved

VII. Comparative Analysis

A. Experiment 1 – Comparison DC Request Service Timing.

The graph shows reduction in average DC service request timing for proposed algorithm as compared to service proximity service broker algorithm.

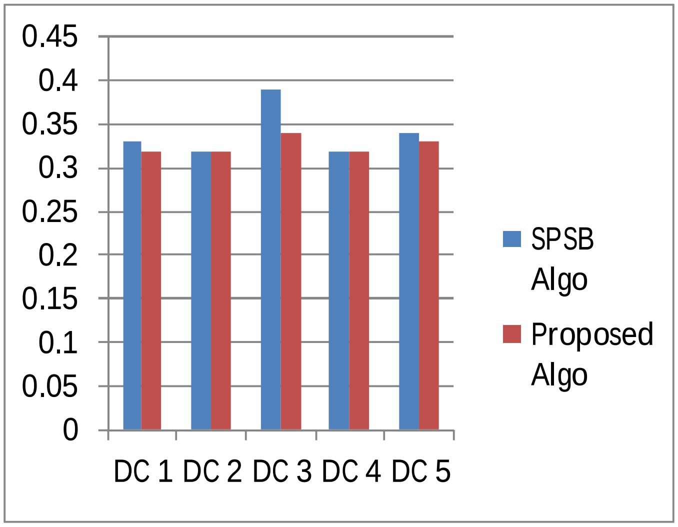

Figure 10.Graph Showing DC Request Service Timing

B. Experiment 2 – Comparison of DC Costing.

The graph shows reduction in average DC Costing timing for proposed algorithm as compared to service proximity service broker algorithm.

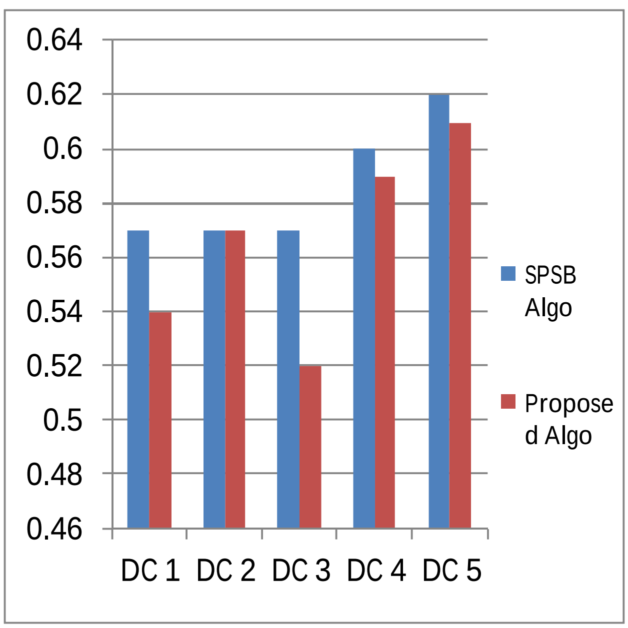

Figure 10.Graph Showing DC Costing

© 2014, IJCSMC All Rights Reserved

VIII. Conclusion

Considering the unique features of long-connectivity applications, an algorithm is proposed which improves the performance of existing service proximity service broker algorithm. Proposed algorithm optimizes the number of connections and reduces average DC request service timing. Finally, experiments show that proposed algorithm improves the performance of existing SPSB algorithm. The future work may include desig and development of effective service broker algorithm for multimedia and live streaming web applications.

References

[1] Vogels, "A head in the clouds the power of infrastructureas as service", Proceedings of the 1st Workshop on Cloud Computing and Applications, CCA'08.
[2] Armbrust, A.Fox, R.Griffith, A.Joseph, R.Katz, A.Konwinski, G.Lee,DPatterson, A.Rabkin, I.Stoica, , "Above the clouds : A Berkeley view of cloud computing, Technical Report UCB / EECS -2009-28,2009.
[3] J.Carolan, S.Gaede, J.Baty, G.Brunette, A.Licht, J.Remmell, L.Tucker, J.Weise, "Introduction to cloud computing architecture", whitepaper, 2009.
[4] Rimal B.P., Choi E. and Lumb I.**, "**A Taxonomy and Survey of Cloud Computing Systems**"**, IEEE 5th International Joint Conference on INC, IMS and IDC, pp - 44-51, 2009.
[5] Mata-Toledo R. and Gupta P. (2010)**, "**Cloud Load Balancing Techniques: A Step towards Green**"** Journal of Technology Research, Volume No-2 issue - 1, 1-8.
[6] Alakeel A. M., "A Fuzzy Dynamic Load Balancing Algorithm for Homogenous Distributed Systems" International Journal of Computer Science and Network Security, 10(6), 153-160, 2010.
[7] E.Marinelli, "Hyrax: cloud computing on mobile devices using Map Reduce", Master's Thesis, Carnegie Mellon University, 2009.
[8] M.Satyanarayanan, P.Bahl, R.Caceres, N.Davies, "The case for VM-based cloudlets in mobile computing, IEEE Pervasive Computing, volume – 8, 2009, pp - 14–23.
[9] Mei,W., Chan,T. Tse,A , "Tale of clouds : paradigm comparis on sandsome thoughtson research issues", ProceedingsoftheAsia-PacificServices Computing Conference, APSCC' IEEE, 2008, pp - 464–469.
[10] Xiaona Ren, Rongheng Lin,Hua Zou, "a dynamic load balancing strategy for cloud computing platform based on exponential smoothing forecast", IEEE ccis2011, pp – 220 – 225, 2011.
[11] Qingfeng Liu, Xie Jian, Jicheng Hu, "An optimized solution for mobile environment using mobile cloud computing", IEEE International Conference on Wireless Communications, Networking and Mobile Computing , 2009
[12] Qi Cao, Zhi-Bo Wei, Wen-Mao Gong, "An Optimized Algorithm for Task Scheduling Based On Activity Based Costing in Cloud Computing", International Conference on Bioinformatics and Biomedical Engineering , 2009, Page(s): 1 – 3.
[13] Luo Yongjun, LiXiaole, and Sun Ruxiang, "Load Balancing Algorithms Overview," Information Development and Economy, vol. 18, 2008, pp. 134–136.
[14] M. Randles, D. Lamb, and A. Taleb-Bendiab, "A Comparative Study into Distributed Load Balancing Algorithms for Cloud Computing," 2010 IEEE 24th International Conference on Advanced Information Networking and Applications Workshops, 2010, pp. 551–556.
[15] Tian Shaoliang, Zuo Ming, and Wu Shaowei, "An improved load balancing algorithm based on dynamic feedback," Computer Engineering and Design, vol. 28, 2007, pp. 572–573.
[16] Zheng Qi, "Load balancing algorithm based on dynamic feedback," Computer Age, 2006, pp. 49–51.
[17] Xiaona Ren ; Rongheng Lin ; Hua Zou, " A dynamic load balancing strategy for cloud computing platform based on exponential smoothing forecast", 2011 IEEE International Conference on Cloud Computing and Intelligence Systems (CCIS), Page(s): 220 – 224
[18] Elghoneimy, E.; Bouhali, O.; Alnuweiri, H., "Resource allocation and scheduling in cloud computing", International Conference on Computing, Networking and Communications (ICNC), 2012 , Page(s): 309 – 314.
[19] Khazaei H., Misic J. "Performance Analysis of Cloud Computing Centers Using M/G/m/m+r Queuing Systems", IEEE Transactions on parallel and distributed systems, Volume: 23, Issue : 5,2012, PP: 936 – 943.
[20] Murata Y., Egawa R., Higashida M., Kobayashi H.,"A History-Based Job Scheduling Mechanism for the Vector Computing Cloud" , IEEE/IPSJ International Symposium on Applications and the Internet(SAIT), 2010, Page(s): 125 – 128.
[21] Chih-Yung Chen; Hsiang-Yi Tseng, "An Exploration of the Optimization of Excutive Scheduling in the Cloud Computing" , IEEE International Conference on Advanced Information Networking and Applications Workshops (WAINA), Page(s): 1316 – 1319,2012.
[22] Hu Wu; Zhuo Tang; Renfa Li , "A priority constrained scheduling strategy of multiple workflows for cloud computing", IEEE international conference on Advanced Communication Technology (ICACT), Page(s): 1086 – 1089, 2012.

© 2014, IJCSMC All Rights Reserved

[23] Chih-Yung Chen; Hsiang-Yi Tseng, “An Exploration of the Optimization of Excutive Scheduling in the Cloud Computing” , IEEE International Conference on Advanced Information Networking and Applications Workshops (WAINA), Page(s): 1316 – 1319,2012.

[24] Gotoda, S.; Ito, M.; Shibata, N., “Task Scheduling Algorithm for Multicore Processor System for Minimizing Recovery Time in Case of Single Node Fault” , IEEE/ACM International Symposium Cluster, Cloud and Grid Computing (CCGrid), Page(s): 260 – 267, 2012

[25] Deepak Kapgate et. al., “predictive load balancing strategy for reduction of latency in mobile cloud computing”, IJCCER, Vol No - ,Issue No - ,2013

[26] Bhathiya Wickremasinghe CloudAnalyst: A CloudSim-based Tool for Modelling and Analysis of Large Scale Cloud Computing Environments MEDC Project. 433-659 distributed computing project, csse dept., university of melbourne June 2009.

[27] Bhathiya Wickremasinghe, Rodrigo N. Calheiros, and Rajkumar Buyya. CloudAnalyst: A CloudSim-based Visual Modeller for Analysing Cloud Computing Environments and Application. The Cloud Computing and Distributed Systems (CLOUDS) Laboratory, The University of Melbourne, Australia, 2009.

© 2014, IJCSMC All Rights Reserved