American-Eurasian Journal of Scientific Research 11 (1): 38-44, 2016
ISSN 1818-6785

DOI: 10.5829/idosi.aejsr.2016.11.1.22823

# Prioritizing Service Based on Multi-Tier Service Approximation Technique for Dynamic Selection and Composition of Cloud Services in Multi Tier Clouds

[1]*G.R. Anantha Raman and* [2]*S. Sakthivel*

[1]Adhiyamaan College of Engineering. Hosur, Research Scholar, Anna University, Chennai, India
[2]Sona College of Technology, Salem, Research Coordinator, Anna University, Chennai, India

**Abstract:** The increasing amount of information and the growth of information technology paved the way for internet users to perform their desired task without spending more on the resources using cloud computing. The cloud provides various services at different level of architecture like service layer, platform layer and database layer. To access these services the users has to be registered and could access the desired information through appropriate service provided by the service provider. There will be more than one service available to perform specific task and the service itself may require a different level service. The throughput of the environment is highly depending on how many service requests has been fulfilled on any particular time. Even though there exist number of service the service selection makes the difference and there are many approaches have been discussed earlier to perform service selection, but suffer with the problem of unavailability, poor completion status, time complexity and latency. To overcome these issues, we propose a multi tier service based service prioritization scheme to perform service selection. The selected services will be bundled to produce a composite service which will be given to the user request to complete the task. The service approximation technique computes service depthless measure and service throughput measure. Based on these two, we compute the service trust weight to prioritize the services. A top weighted service will be selected to fulfill the service request of the user and reduces the latency and time complexity of service selection. The proposed approach increases the throughput of overall environment and reduces the time complexity.

**Key words:** Multi Tier Clouds · Service Prioritization · Composite Services · Service Approximation

## INTRODUCTION

The development of information technology has opened a big gate for the information surfers through cloud environment. The cloud is a platform or environment where the user can store their information and share them between various users. Similarly the resource shared may be of anything like files, memory devices, processing elements or anything. Generally there are an organization which has huge amount of information has to be stored or they maintain customer information of millions which needs vast amount of storage medium and cost more. The scarcity of memory and the cost factor affects the organization to maintain their information as needed. Also there are situations that a user has a batch of job which needs more memory and higher processing elements and could not be afford by them. To resolve the situation like this, there are organizations like IBM provides the cloud where the user can maintain the data by paying marginal amount which is negligible. What the cloud organizations do is they provide mass storage and processing elements which can be accessed by the users of the network or cloud.

The resources provided by the cloud organizations could not be accessed directly to provide data integrity, security and audit ability. The user has to register the environment and he will be verified by the third party auditor before he allowed accessing the data or resource. To access the cloud resource the environment provides various services at different levels. For example in application layer, they provide Software as a Service (SaaS), in the physical layer they provide Platform as a Service (PaaS) and for the data layer Data as a Service (DaaS) is being provided. Whatever the resource the user

**Corresponding Author:** G.R. Anantha Raman, Adhiyamaan College of Engineering. Hosur, Research Scholar, Anna University, Chennai, India.

has to access they can access only through these services. To get more efficient results, the service selection has to be performed in efficient way so that the quality of service could be improved.

The service prioritization is the process of prioritizing the services based on different measures like service availability, service completeness, service efficiency and many more. From available similar service an appropriate service has to be selected. Various service providers may provide same set of services but differs with the quality of completeness and other qualities of service. Also there are situations where some services need the support of other services which has to be composed to form composite service. This situation highly requires the condition of choosing exact service to bundle a composite service.

By reviewing all these the services has to be prioritized and selected based on various strategies to generate a composite service. The composite service is the collection of set of references grouped to complete a specific task. The selection of service has to be approximated that each service has to be verified for its trustworthy, leakage of information and availability and completeness. The service approximation could be used to prioritize the service and allow the service request to be completed in efficient manner.

**Related Works:** There are many approaches has been discussed in literature for efficient cloud computing in multi clouds. We discuss few of them here in this chapter.

Intelligent Cloud Service Selection Using Agents [1], propose a new technique performance factor for the provision of services based on intelligence. The research objective is to enable cloud users in selecting cloud service according to their own requirements. The technique assigns performance factor for each service provided by cloud and ranks it as whole. By doing so, quality of the services can be highly improved. We validate our approach with a case study, which emphasizes the need to rank cloud services of widely spreading and complex domains.

Managing Imprecise Criteria in Cloud Service Ranking with a Fuzzy Multi-criteria Decision Making Method [2-5], discuss an alternative classification of metrics used for ranking cloud services based on their level of fuzziness and present an approach that allows cloud service evaluation based on a heterogeneous model of service characteristics. Our approach allows the multi-objective assessment of cloud services in a unified way, taking into account precise and imprecise metrics. We use fuzzy numbers to model the imprecise service characteristics and vague user preferences and we validate a fuzzy AHP approach that solves the problem of service ranking.

Top-k Web Service Compositions using Fuzzy Dominance Relationship [10], present an approach to automatically compose Data Web services while taking into account the user preferences. User preferences are modeled thanks to fuzzy sets. We use an RDF query rewriting algorithm to determine the relevant services. The fuzzy constraints of the relevant services are matched to those of the query using a set of matching methods. We rank-order services using a justification of Pareto dominance, then compute the top-k service compositions. We propose also a method to improve the diversity of returned compositions while maintaining as possible the compositions with the highest scores. Finally, we present a thorough experimental study of our approach.

Service Selection based on Personalized Preference and Trade-Offs among QoS [6-11], presents a novel service selection method, in which users can represent their non-functional requirements using linguistic terms and use their personalized trade-off strategies for service selection. The method employs fuzzy propositions in conjunction with QoS and price information to compute satisfaction degree of individual non-functional requirements. Based on the personalized trade-off strategy of a user, it then computes the overall satisfaction degree using fuzzy connective operators. The overall satisfaction degree of services is then used to rank services with equivalent functionalities and top-ranked services are recommended. An application example is presented to show the method's effectiveness.

Data center selection based on euro-fuzzy inference systems in cloud computing environments [12-18], proposes a neuro-fuzzy inference-based prediction scheme to select one of multiple data centers in accordance with application workloads. This scheme is used to aggressively capture the time-varying load of data centers by learning and predicting the availability of resources therein. Therefore, it predicts not only the present load but also the future load of data centers in the process of determining a suitable data center. By an autonomic control for data center selection, our scheme can also provide load balancing between data centers.

Energy Efficient Cloud Data Center Management Based on Fuzzy Multi Criteria Decision Making [19], propose and evaluate an approach for power and performance management through intelligent decisions over hotspot detection and migration time of VMs across heterogeneous PMs for mitigation of hotspot of a cloud computing data center. We used two dynamic threshold levels (peak and off-peak) load strategy to implement our decisions. The focus of this paper is to present a new method to find overloaded nodes at upper level threshold (peak load) and also perform PMs consolidation at lower-level threshold and putting idle server to sleep state, by using Best Fit Decreasing based on TOPSISone of the most efficient Multi Criteria Decision Making techniques.

On the Characterization of the Structural Robustness of Data Center Networks [4], analyze the robustness of the state-of-the-art DCNs. First, they present multi-layered graph modeling of various DCNs and they study the classical robustness metrics considering various failure scenarios to perform a comparative analysis; finally present the inadequacy of the classical network robustness metrics to appropriately evaluate the DCN robustness and proposes new procedures to quantify the DCN robustness. Currently, there is no detailed study available centering the DCN robustness.

All the above discussed methods has the problem of mining the service availability and selecting the appropriate service when there is a demand for the service. We propose a naval on demand service selection which is performed on the fly in this paper.

**Proposed Method:** The proposed service approximation based service prioritizing scheme has various functional modules namely Service Approximation, Service Prioritization and service selection and composition. We discuss each of the detain in this chapter.

**Service Approximation:** Whenever the user request arrives the proposed method performs service approximation technique. The method identifies similar set of services which are requested and for each service we compute service depthness measure using number of internal services required to fulfill the request. Also we compute the service throughput measure based on the completion of each service and using both the measure we compute the service weight. The service depthness measure considers various factors like number of times the service has been accessed, their completeness status, number of other service it requires to complete the specific task and so on. The service throughput measure is computed using number of times the service been selected and number of success completion and so on.

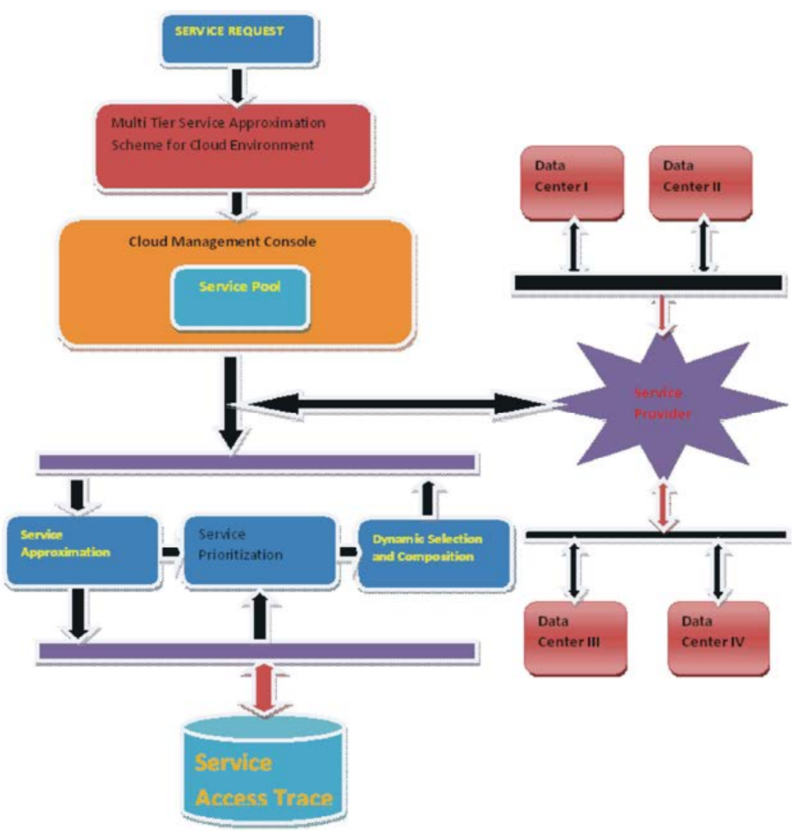



Fig. 1: Proposed System architecture.

**Algorithm:**
Input: Service History Sh, Service Request SR
Output: Service Approximation value set SAVS.
Step1: Identify type of service St = $\int_{i=1}^{N} Service.Type \in (SR.serviceType)$
Step2: Identify set of all similar service SSS.
SSS = $\int_{i=1}^{N} \sum Service.Type \in Sh$
Step3: for each service Si from SSS
Compute service depthness measure SDM.
SDM = $\int \frac{\sum_{i=1}^{N} No\ of\ services\ accessed}{Total\ number\ of\ access.}$
Compute service throughput ST = $\int \frac{Number\ of\ service\ completion}{Total\ number\ of\ access} \times 100$
End.
Step4: for each service si
Compute service approximation value sav = $\int \sum si(SAVS) + (SDM * ST).$
End
Step5: stop.

**Service Prioritization:** The service prioritization is the process of prioritizing the service based on various factors like availability, timeliness and efficiency of the specific service. The service priority value is compute using the above said factors which are retrieved from the service history. The service history has the details about the service claim and their assignment and so on. Based on the details of the claim and assignment factors we compute the service availability and the delay occurred in the service completion. While considering the multi user multi tier multi service environment the efficiency of the system is highly depend on the factors like time complexity and the latency must be less and user must get the feeling that the service is processed in short time and so on.

**Algorithm:**
Input: Service History Sh, Service Approximation value set SAVS
Output: Service priority set Sps.
Step1: initialize service priority set Sps.
Step2: for each service si from SAVS
Compute number of service request araised.
Nosr = $\int \sum SAVS(Sid) \in Sh$
Compute number of assignment.
NoA = $\int \frac{\sum (Si \in \in sh) == True}{SAVS(Sid) \in Sh}$
Compute service priority sp = $\frac{NoA}{Nosr} \times 100$
End
Step3: stop.

**Dynamic Selection and Composition:** The process first identifies set of services necessary to complete the service request. For each service identified the approach computes service approximation and service priority. From computed priority and service approximation value, we compute the service weight which is used to choose the efficient service from the resource pool. The service selection is performed in this manner for each required service and finally the selected services are composed to form the composite service to produce the result.

**Algorithm:**
Input: Service Approximation set SAS, Service priority set SPs.
Output: Composite service CS.

Step1: initialize CS.
Step2: for each service si from SAS
Compute service weight sw = SAS(is)*SPS(is)
End
Step3: Choose service maximum service weight
SL = $\int \sum Service(\max(Sw))$
Step4: Compose selected service from Sl.
Step5:stop.

## RESULTS AND DISCUSSION

The proposed service prioritization scheme based on service approximation has been implemented and tested for its efficiency. The proposed approach has produced efficient results in all the factors of quality of service of cloud computing in multi clouds.

The proposed solution has been implemented Hadoop, which is the cloud computing platform integrated with the proposed solution to evaluate the proposed methodology. We have created three different clouds, each running on different locations and three service providers which are running at N-Number of locations. The proposed method has maintains various data centers and access traces to evaluate the performance of the proposed approach.

To evaluate the performance of the proposed solution, we have measured the following metrics namely, Availability Ratio, Completeness Ratio, Overall Performance Ratio. Availability is the ratio of total requests submitted and total requests handled. The completeness is the ratio between total submitted request and number of requests processed successfully. Overall performance is the ratio of total number requests submitted and number of jobs completed in a particular time frame.

Table 1: Shows the comparison results of service availability

| | Availability Ratio% | | | | |
|---|---|---|---|---|---|
| No of. Users | Tabu | Neurofuzzy | ML-Graph | ODOF | MTSA |
| 1 million | 72 | 82 | 89 | 98.7 | 99.3 |
| 2 million | 69 | 79 | 85 | 97.6 | 98.7 |
| 3 million | 67 | 75 | 81 | 96.6 | 97.6 |
| 5 million | 63 | 72 | 77 | 95.8 | 96.6 |
| 10 million | 59 | 69 | 72 | 95.1 | 95.8 |

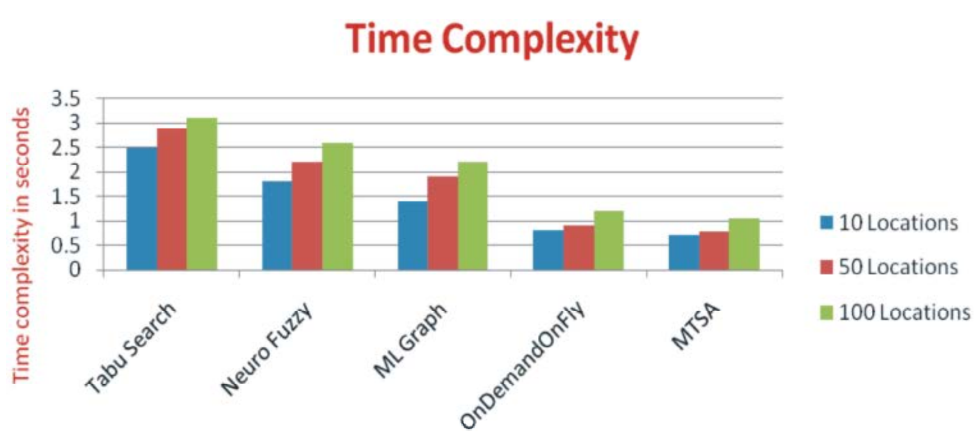
Table 1 shows the service availability ratio of four different algorithms and it shows that the proposed method provides more service availability than other methods.

The Graph 1 shows the time complexity of different methods to access the service where the service and data is available at different locations in the region.

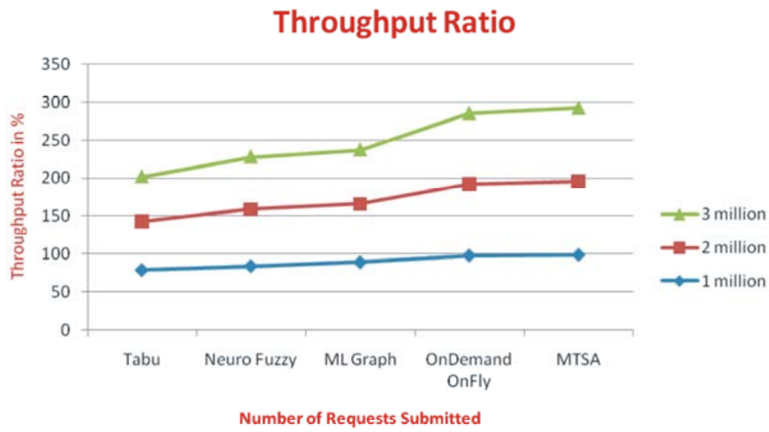
The time complexity is □ (N×M), where N- is the number of locations where the service is available and M- is the number of service providers available. The overall time complexity is computed as follows:

Time complexity Tc = N × Log (M).

The Graph1 shows that the proposed method has produced higher efficient results compare to other algorithms.




Graph 1: Shows the time complexity of different approaches.




Graph 2: Shows the comparison of throughput ratio.

The Graph 2 shows the comparison of overall throughput generated by different algorithms, it shows that the proposed method has produced higher throughput than other methods.

## CONCLUSION

We proposed a new dynamic prioritization scheme based on multi tier service approximation technique identifies set of all services necessary to complete the task. From the identified required services a set of available service are identified and for each service we compute the service approximation and service priority value using the service depthness and service throughput values. Base on computed measure we compute the service weight for each service. Finally a set of service will be selected based on weight and composed to provide final composite service. The proposed approach has produced higher efficient results in all the factors and reduces the time complexity also.

## REFERENCES

1. Imran Mujjadhid Rabbani, 2013. Intelligent Cloud Service Selection Using Agents, Springer, Advances in Intelligent Systems and Computing, 209: 105-114.
2. Myougnjin, K., L. Hanku, Y. Hyogun, K. Jee-In and K. HyungSeok, 2011. IMAV: An Intelligent Multi-Agent Model Based on Cloud Computing for Resource Virtualization. In: 2011 International Conference on Information and Electronics Engineering, IPCSIT, IACSIT Press, Singapore, 6: 199-203.
3. Shailesh, K., 2011. Chandramohan: Personalized Web Service Selection, International Journal of Web & Semantic Technology (IJWesT), 2(2): 78-93.
4. Singh, A. and M. Malhotra, 2012. Agent Based Framework for Scalability in Cloud Computing, International Journal of Computer Science & Engineering Technology (IJCSET) , 3(4): 41-45.
5. Ioannis Patiniotakis, Stamatia Rizou, Yiannis Verginadis and Gregoris Mentzas, 2013. Managing Imprecise Criteria in Cloud Service Ranking with a Fuzzy Multi-criteria Decision Making Method, springer, Service Oriented and Computing, 8135: 34-48.
6. Garg, S.K., S. Versteeg and R. Buyya, 2011. SMICloud: A Framework for Comparing and Ranking Cloud Services. Presented at the Fourth IEEE International Conference on Utility and Cloud Computing, Victoria, NSW, pp: 210-218.
7. Chan, K.Y., T.S. Dillon and C.K. Kwong, 2012. An Enhanced Fuzzy AHP Method with Extent Analysis for Determining Importance of Customer Requirements. In: Chan, K.Y., Kwong, C.K., Dillon, T.S. (eds.) Comput. Intell. Techniques for New Product Design. SCI, Springer, Heidelberg, 403: 79-94.
8. Pawluk, P., B. Simmons, M. Smit, M. Litoiu and S. Mankovski, 2012. Introducing STRATOS: A Cloud Broker Service. In: 5th IEEE International Conference on Cloud Computing (CLOUD), pp: 891-898.
9. Almulla, M., K. Almatori and H. Yahyaoui, 2011. A QoS-based Fuzzy Model for Ranking Real WorldWeb Services. Presented at the IEEE International Conference on Web Services.
10. Benouaret, K., D. Benslimane, A. Hadjali and M. Barhamgi, 2011. Top-k Web Service Compositions using Fuzzy Dominance Relationship. Presented at the IEEE International Conference on Services Computing.

11. Liu, X(F.), K.K. Fletcher and M. Tang, 2012. Service Selection based on Perso-nalized Preference and Trade-Offs among QoS. Presented at the IEEE First International Conference on Service Economics.
12. Nepal, S., W. Sherchan, J. Hunklinger and A. Bouguettaya, 2010. A Fuzzy Trust Management Framework for Service Web. Presented at the IEEE International Conference on Web Services.
13. Jenn Wei Lin, 2013. QoS-Aware Data Replication for Data-Intensive Applications in Cloud Computing Systems, IEEE Transactions on cloud computing, 1(1): 1.
14. Doyle, J., 2013. Stratus: Load Balancing the Cloud for Carbon Emissions Control, IEEE Transactions on Cloud Computing, 1(1): 1.
15. Zhidongzen cloud computing based on trusted computing platform, International conference on intelligent computation technology and automation, 2010.
16. Kassakian, J.G. and R. Schmalensee, 2011. (Study Co-chairs), the future of the electric grid: An interdisciplinary MIT study, tech. Rep., Massachusetts Institute of Tech.
17. Shao Shenghan, 2013. Development of physical-based demand response-enabled residential load models, IEEE Transaction on Power Systems, 28(2): 607-614.
18. Joon Min Jil, 2013. Data center selection based on neuro-fuzzy inference systems in cloud computing environments, Springer, The Journal of Supercomputing, 66(3): 1194-1214.
19. Jing ShahzadAlip and Siyuan Kun, 2013. She Energy Efficient Cloud Data Center Management Based on Fuzzy Multi Criteria Decision Making, Springer, Find out how to access preview-only contentProceedings of the Sixth International Conference on Management Science and Engineering Management Lecture, Notes in Electrical Engineering, 185: 355-370.