ISSN (ONLINE): 2454-9762
ISSN (PRINT): 2454-9762
Available online at www.ijarmate.com

International Journal of Advanced Research in Management, Architecture, Technology and Engineering (IJARMATE)
Vol. 2, Issue 4, April 2016

Recommending QOS Web Service Based On Location with Agglomerative Hierarchical Clustering

Thamizhazhagi.M[1], Umamaheswari.B[2]

PG Scholar, Department of CSE, Vandayar Engineering College, Thanjavur, India[1]

Assistant Professor, Department of CSE, Vandayar Engineering College, Thanjavur, India [2]

***Abstract*— Recommendation techniques aim to support users in their decision-making while the users interact with large information spaces. Recommendation has been a hot research topic with the rapid growth of information. In the field of services computing and cloud computing, efficient and effective recommendation techniques are critical in helping designers and developers analyze the available information intelligently for better application design and development. To recommend Web services that best fit a user's need, QoS values which characterize the non-functional properties of those candidate services are in demand the QoS information of Web service is not easy to obtain, because only limited historical past invocation records exist. So in this paper present a model named CLUS for reliability prediction of atomic Web services, which estimates the reliability for an ongoing service invocation based on the data merged from previous invocations. Then aggregates the past invocation data using agglomerative hierarchical clustering algorithm to achieve better scalability comparing with other recent approaches. In addition, the proposes a model-based collaborative filtering approach to recommending the web services based user location and reviews. The experimental results provide less error rate at the time of recommendation..**

***Index Terms*— QoS, Recommendation, Clustering, Agglomerative hierarchical clustering, Prediction.**

I. INTRODUCTION

Web service has been an important solution to achieve resource sharing and application integration in the Internet era, which can develop the most promising e-commerce software featured with the on-demand changing computing paradigm, through service reuse and dynamic synthesis. Under the interoperability of Web service, complex business interactions fulfilled by the process of dynamic discovery, integration, coordination and execution of distributed service entities (atomic or composite service), via published and discoverable interfaces, can improve the productivity and work quality of enterprise. However, due to the heterogeneous, open and collaborative nature of Internet, any application failure may immediately result in service interrupt, which will seriously impact the correctness and reliability of core business process of service-oriented software. Although several languages and technologies have been proposed, e.g., BPEL4WS, WS-CDL, WSCL, and OWL-S, they only concern on Orchestration and Choreography composition behaviors for manufacturing a correct Web service at design phase. The functional requirement of business logics is their prior job in absent of providing related strategies or mechanisms to dynamically monitor both functional and nonfunctional attributes of Web service at runtime phase. But the nonfunctional property impacted by random behaviors commonly refers to the quality aspect of Web service, such as security, reliability and availability. Consequently, how to effectively monitor service behaviors remains to be a big challenge in the area of Web service research. An enormous amount of data is generated by various service-oriented systems and cloud applications day by day. It is challenging for users to find the right information they want from such large volume of data. For example, determining the optimal Web service from a large number of service candidates when making service selection, detecting vulnerable components in complex service oriented applications and cloud applications, identifying suitable servers for deploying cloud applications, and so on. In the field of services computing and cloud computing, efficient and effective recommendation techniques are critical in helping designers and developers analyze the available information intelligently for better application design and development. Collaborative filtering methods are widely adopted in recommender systems. A memory-based approach is one type of the most widely studied collaborative filtering approaches. The most analyzed examples of memory-based collaborative filtering include user-based approaches, item-based approaches, and their fusion. User-based and item-based approaches often use the vector similarity method and the PCC method as the similarity computation methods. Compared with vector similarity, PCC considers the differences in the user rating style when calculating the similarity. The rating-based collaborative filtering approaches try to predict the missing QoS values in the user-item matrix as accurately as possible. However, in the ranking-oriented scenarios, accurate missing value prediction may not lead to accuracy ranking prediction. Therefore, ranking-oriented collaborative filtering approaches are

All Rights Reserved © 2016 IJARMATE

ISSN (ONLINE): 2454-9762
ISSN (PRINT): 2454-9762
Available online at www.ijarmate.com

International Journal of Advanced Research in Management, Architecture, Technology and Engineering (IJARMATE)
Vol. 2, Issue 4, April 2016

becoming more attractive. The existing collaborative filtering-based approaches have demonstrated promising results they suffer from potential scalability and accuracy issues. Specifically, these approaches require storing values for every observed service-user pair. Using k-means algorithm cluster the user past invocation. Similarly, the success of a service invocation depends on a variety of other factors that have previously been either only implicitly considered or not considered at all, which may hamper the accuracy of the predictions.

II. Related Works

The successful approaches for prediction of atomic service reliability are based on the collaborative filtering technique.

A. *Zheng .Z small et al(2)*

Designing effective and accurate reliability prediction approaches for the service-oriented systems has become an important research issue. In this paper, we propose a collaborative reliability prediction approach, which employs the past failure data of other similar users to predict the Web service reliability for the current user, without requiring real-world Web service invocations. A user-collaborative failure data sharing mechanism and a reliability composition model for the service-oriented systems. However, without comprehensive evaluation, we cannot collect sufficient past failure data of the Web service components. It is difficult for the system designer to determine whether the service-oriented system is reliable enough for release. Software reliability prediction is a task to determine future reliability of software systems based on the past failure data. This proposed for predicting software reliability by observing, accumulating, and analyzing previous failure data. Traditionally, comprehensive testing schemes are conducted on the software systems to collect failure data and to make sure that the reliability threshold has been achieved before releasing the software to the customers or end users.

The reliability of a service-oriented system not only relies on the system itself, but also heavily depends on the remote Web services and the unpredictable Internet. Service-oriented architecture (SOA) is becoming a major software framework for building complex distributed systems. Reliability of the service-oriented systems heavily depends on the remote Web services as well as the unpredictable Internet. Designing effective and accurate reliability prediction approaches for the service-oriented systems has become an important research issue. It propose a collaborative reliability prediction approach, which employs the past failure data of other similar users to predict the Web service reliability for the current user, without requiring real-world Web service invocations. Large-scale real-world experiments are conducted.

A collaborative framework is proposed for predicting reliability of service-oriented systems. Different from previous reliability prediction approaches, our approach employs past failure data of similar service users for making reliability prediction for the current service user. Extensive experiments are conducted using real-world Web service dataset, which contains 1.5 millions real-world Web service invocation results from150 distributed service users on 100 real-world Web services.

B. *Coppolino .L small et al (5)*

The formal approach that allows a workflow architect to perform reliability analysis of a SOA-based service. The approach exploits the concept of reliability patterns to derive an aggregate reliability function and it is suited for a wide class of workflow processes. The approach is implemented in a tool (more precisely, as a plug-in for Active BPEL Designer). The tool allows the system architect to evaluate the impact—in terms of reliability—of possible workflow alternatives, as early as in the first steps of the design. Providing reliable compositions of Web Services is a challenging issue since the workflow architect often has only a limited control over the reliability of the composed services.

The architect can instead achieve reliability by properly planning the workflow architecture. To this end he must be able to evaluate and compare the reliability of multiple architectural solutions. In this present a useful tool which allows to conduct reliability analysis on planned workflows, as well as to compare the reliability of alternative solutions in a what-if analysis. The tool is implemented as a plug-in for the widely adopted Active BPEL Designer and exploits the concept of reliability pattern to evaluate the reliability formula of the workflow. The effectiveness of the approach and the operation of the tool are demonstrated with respect to a case study of a business security infrastructure realized by orchestrating simple security services.

The main limitation of this approach lies in the possibility of getting the reliability expression only for those workflows that can be obtained by composing the patterns described. To overcome such limitation, It start from results presented in where the authors present, by extending results reported , a set of 43 workflow patterns whose combinations can provide pretty every workflow. Starting from this set of patterns, then it identifies the combinations of them that are meaningful from a reliability point of view and derive for them a reliability expression. It refers such combinations as “reliability patterns”. Since virtually any workflow can be obtained by combining the workflow patterns, “reliability patterns” can also be applied to obtain the reliability formula for any workflow. It demonstrate our approach to the patterns defined in, similarly it is possible to derive new reliability patterns from the remaining patterns defined By doing so it verify that, not only all the patterns defined are also obtained as reliability patterns, but new patterns, not considered, such as the “Multi-Merge Parallel”, are also identified and considered for reliability evaluation.

C. *Baresi .L small et al(3)*

The service paradigm, together with virtualization technology, is imposing a profound rethinking of many

All Rights Reserved © 2016 IJARMATE

ISSN (ONLINE): 2454-9762
ISSN (PRINT): 2454-9762
Available online at www.ijarmate.com

International Journal of Advanced Research in Management, Architecture, Technology and Engineering (IJARMATE)
Vol. 2, Issue 4, April 2016

complex software systems. Providing virtual infrastructures as services is gaining more and more momentum, and is imposing a more cohesive view of the different layers that constitute a software system. Applications, service platforms, and virtualized infrastructures have become tightly integrated. It is now possible to change software's quality of service (QoS) by changing the configuration of the virtual machines it uses. It allows us to define how to collect, aggregate, and analyze runtime data in a multilayered system.

A framework for event correlation and aggregation that supports and provides a dashboard for on-line and off-line drill-down analyses of collected data. If the designer specifies a property, the calculation of the indicator will only take into account the requests that satisfy that property. In this case the property can make use of an implicit alias called "input" to refer to the contents of each collected request message. Throughput outputs the number of serviced requests, and reliability outputs the number of correct outputs over the total number of requests. The correctness of an output is determined by the fact that it satisfies a defined property In this case the property can make use of implicit aliases "input" and "output" to refer to matching request and response messages. This is useful if the response message's correctness depends on the content of the corresponding request message.

The integration focused heavily on BPEL processes, and suffered from a lack of well-devised abstractions that could guide the integration in a more general way. This approach that takes into account the multiple layers that constitutes modern Web applications. They instrument the system across all its layers, and exploit the BPEL workflows that compose the services to generate structural cross-domain dependency graphs. Then they monitor all the components and correlate their metrics to present the system manager with a comprehensive multi-layer diagnosis. Although similar to our work in scope, our focus is on a general-purpose multi-layer monitoring language, and on how to trigger data correlations at different layers, instead of on how to generate dependency graphs from BPEL specification. Empirical assessment shows that the impact of the approach on runtime performance is negligible. They can even instantiate new virtual machines to address load problems, or move our software from one infrastructure to another if its quality is not acceptable. Even if one might say that this is nothing new, the key distinctive feature is the ease with which the different parameters can be changed, and the different runtime executors (e.g., virtual machines) added or modified to impact a system's behavior.

D. Yu .Q small et al(8)

The collaborative filtering algorithm for making QoS-aware service recommendations. The proposed algorithm effectively tackles the QoS data sparsity issue, which leads to accurate QoS predictions. In this regard, it can be modeled as a general matrix completion problem, which is to recover a large QoS matrix from a small subset of QoS entire's. The infinite number of ways to complete an arbitrary QoS matrix makes the problem extremely ill posed. The highly sparse QoS data further complicates the challenges. Nonetheless, real-world QoS data exhibits two key features, which can be leveraged for accurate QoS predictions, leading to high-quality service recommendations.

First, QoS delivery can be significantly affected by small number dominant factors in the service environment (e.g., communication link and user-service distance). Hence, it is natural to assume that the QoS matrix has a low-rank or approximately low rank structure. Second, users (or services) that share common environmental factors are expected to receive (or deliver) similar QoS and hence can be grouped together. The proposed approach seamlessly amalgamates these two features into a unified objective function and employs an effective iterative algorithm to approach the optimal completion of an arbitrary QoS matrix.

Service-oriented computing offers an attractive paradigm for the provisioning and consuming of computing resources for a wide spectrum of domains. The large number of applications heavily taking advantage of this new computing paradigm has lead to the deployment of substantial Web services. Many Web services may also offer similar functionalities but vary from each other in terms of the Quality of Service (QoS) they deliver. The QoS is mainly made of user centered quality parameters and examples include availability, response time, and so on. Meanwhile, Web services are autonomous entities as they are developed and deployed by independent developers in an open online environment. Therefore, some services may not deliver on what they promise. Realizing the full potential of service computing requires the ability to efficiently and accurately retrieve services.

A TNR-MF algorithm for accurate QoS prediction. The proposed algorithm incorporates both the low-rank structure and the clustered representation of real-world QoS data. It exploits trace norm regularized matrix factorization to seamlessly amalgamate these two features into a single unified objective function. A novel iterative algorithm is developed that employs simple gradient steps and efficient update rules to recover a QoS matrix from a small subset of observed QoS entire's. Experimental results on two real-world QoS datasets and comparison with both neighborhood and model based collaborative filtering algorithms clearly justify the effectiveness of the proposed algorithm.

III. PROPOSED DESIGN

Quality-of-service can be validating at the server side or at the client side. While server-side QoS properties provide good quality of the cloud service capacity client-side QoS properties provide more realistic measurements of the user usage experience client-side QoS properties include response time, throughput, failure probability, etc. It focuses the scalability and accuracy issues in existing approaches. And avoid ideal recommendation from dissimilar invocation data. Implement model based collaborative filtering with clustering approach to cluster similar services. It propose a two-phase, that is, Web Service Clustering Phase due to the uncertainty of web service data, we can cluster user, service, environment

All Rights Reserved © 2016 IJARMATE

ISSN (ONLINE): 2454-9762
ISSN (PRINT): 2454-9762
Available online at www.ijarmate.com

International Journal of Advanced Research in Management, Architecture, Technology and Engineering (IJARMATE)
Vol. 2, Issue 4, April 2016

specific parameters using agglomerative hierarchical clustering to create two dimensional matrix and combine environment specific parameters to provide three dimension data. Web Service Prediction Phase to achieve the reliability prediction from the variety of historical past invocation data values.

A. *Web Service Creation*

A Web service is a method of communication between two electronic devices over a network. It is a software function provided at a network address over the Web with the service always on as in the concept of utility computing. Web service provides an interface to collect data from heterogeneous environments.

B. *Service Collection*

Web service data is collected from user's access. Data collection is a input component of an information system. Data is collected is in the form of excel data. These datasets are known as historical datasets.

C. *Clustering of Services*

Cluster the datasets which are collected by above module these datasets, predict service specific, user specific and environmental specific similarity. The retrieved QoS values are clustered using agglomerative hierarchical clustering (AHC).And to find the similar services from overall web services. It is useful to eliminate irrelevant and ideal services.

D. *Service Prediction*

Collaborative filtering (CF) is a technique used by some recommender systems. Collaborative filtering has two senses, a narrow one and a more general one. In general, collaborative filtering is the process of filtering for information or patterns using techniques involving collaboration among multiple agents, viewpoints, data sources, etc. From preprocessed data, analyze rating similarity. And compute predictive rating for services. Based on this approach users may discover valuable services quickly.

E. *Recommendation of Services*

Recommender systems or recommendation systems are a subclass of information filtering system that seek to predict the 'rating' or 'preference' or 'similarity' that user would give to an item. This module provides best valued service for recommending services. We can also recommend the web services based on user's current locations. So user gets nearest locations for each service and improves the service quality.

IV. SYSTEM ARCHITECTURE

Clustering is based on QoS parameters which are requested by the user at the searching time. It significantly increases the performance of service discovery by analyzing the best cluster value instead of analyzing the huge amount of data. The proposed model includes the four kinds of steps. They are,

- Filtering the web service based on keywords
- Extracting the QoS values from WSDL document
- Forming the cluster using the extracted QoS values
- Selecting the most suitable web service from the cluster.

Filtering is the very first step in web service clustering. It parses the WSDL document which contains a web service name and other QoS properties for the specific web service and also it has a URL for the web service. Finally it returns the web services Name which is specified by the user. After the filtering process, the web service name will be displayed. The user specified QoS values are retrieved from the WSDL document based on filtered web service names. The retrieved QoS values are clustered using agglomerative hierarchical clustering (AHC). It is a method of cluster analysis which seeks to build a hierarchy of clusters. Strategies for hierarchical clustering generally fall into two types: Agglomerative: This is a "bottom up" approach: each observation starts in its own cluster, and pairs of clusters are merged as one moves up the hierarchy. Divisive: This is a "top down" approach: all observations start in one cluster, and splits are performed recursively as one moves down the hierarchy. In general, the merges and splits are determined in a greedy manner. The results of hierarchical clustering are usually presented in a tree structure. AHC algorithm implemented as clustering algorithm. And to find the similar services from overall web services. And it is useful to eliminate irrelevant and ideal services. Then implement location based collaborative filtering to filter cluster results and recommend services to users.

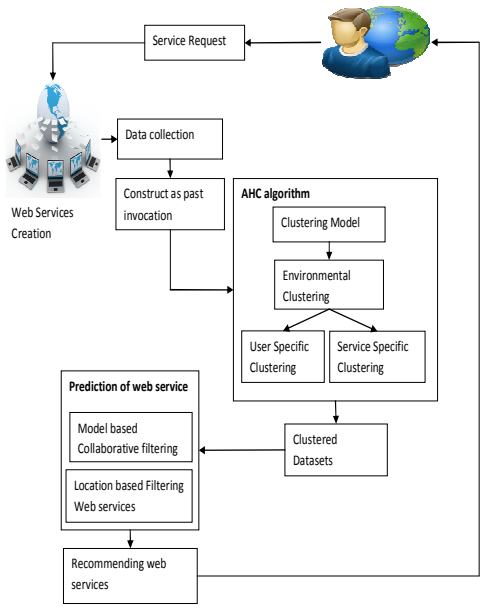

Fig 1: Architecture overview

All Rights Reserved © 2016 IJARMATE

ISSN (ONLINE): 2454-9762
ISSN (PRINT): 2454-9762
Available online at www.ijarmate.com

International Journal of Advanced Research in Management, Architecture, Technology and Engineering (IJARMATE)
Vol. 2, Issue 4, April 2016

V. EXPERIMENTAL ANALYSIS

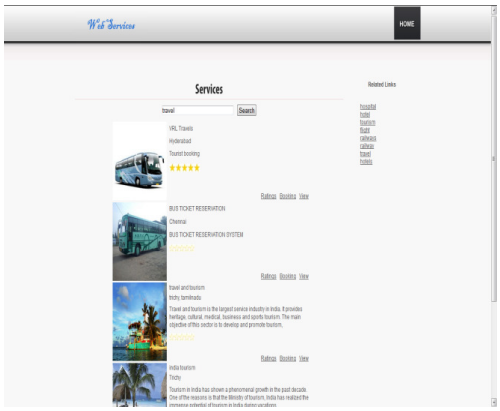

Fig 2: Service collection

Fig3: Service details

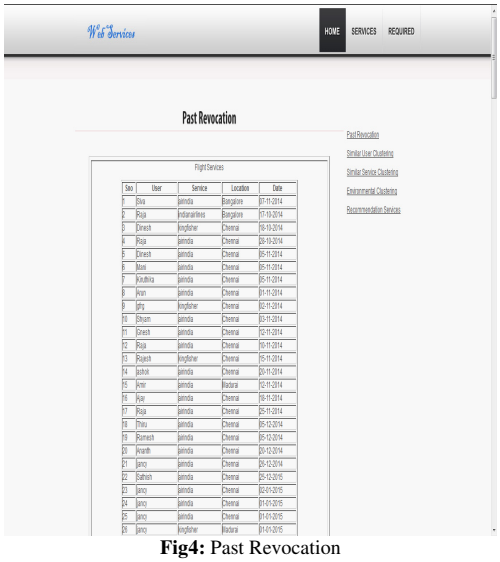

Fig4: Past Revocation

Fig5: Similar user clustering

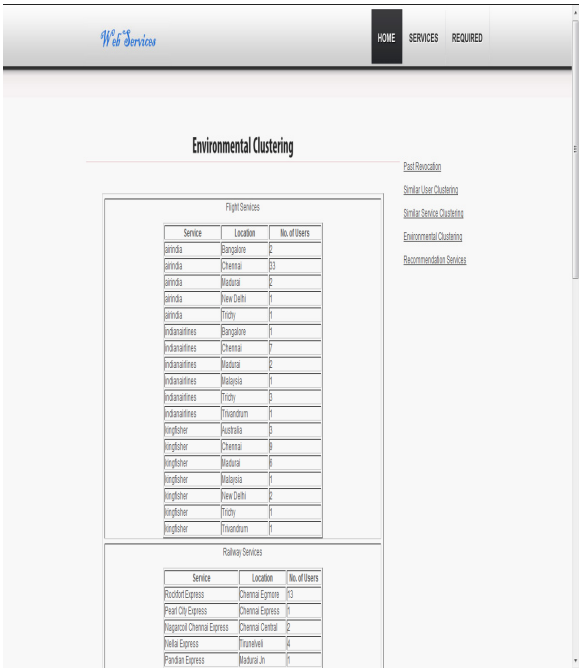

Fig6: Environmental Clustering

VI. CONCLUSION

Web services and cloud computing, an increasing number of services are emerging on the Internet. As a result, service-relevant data become too big to be efficiently processed by traditional approaches. Clustering is done automatically using agglomerative hierarchical clustering. Technically, this approach is enacted around two stages. In the first stage, the available services are divided into small-scale clusters, in logic, for further processing. At the second stage, a collaborative filtering algorithm is imposed on

All Rights Reserved © 2016 IJARMATE

ISSN (ONLINE): 2454-9762
ISSN (PRINT): 2454-9762
Available online at www.ijarmate.com

International Journal of Advanced Research in Management, Architecture, Technology and Engineering (IJARMATE)
Vol. 2, Issue 4, April 2016

one of the clusters. Since the number of the services in a cluster is much less than the total number of the services available on the web, it is expected to reduce the online execution time of collaborative filtering. In view of this challenge, a Clustering-based on location is proposed in future, which aims to find similar services in the cluster to recommend the services.

REFERENCES

[1] T.H.Tan, E.Andre, J.Sun, Y.Liu, J.S.Dong, and M. Chen, "Dynamic synthesis of local time requirement for service composition," in International Conference on Software Engineering,2013.

[2] Z.Zheng and M.R.Lyu, "Colloborative reliability prediction of service oriented systems," in ACM/IEEE International Conference on Software Engineering-Volume 1,ACM,2010.

[3] L.Baresi and S.Guinea,"Event-based multi-level service monitoring," in ICWS, pp.83-90, 2013.

[4] R.Xu and I.Wunsch,D., "Survey of clustering algorithms,"Neural Networks, IEEE Transactions on,2005.

[5] L.Coppolino,L.Romano,and V.Vianello,"Security engineering of soa applications via reliability patterns.,"JSEA,2011.

[6] X.Su and T.M.Khoshgoftaar,"A Survey of collaborative filtering techniques," Adv.in Artif.Intell., vol.2009.

[7] M.R.Lyu,ed.,Handbook of software reliability engineering. McGraw-Hill, Inc., 1996.

[8] Q.Yu,Z.Zheng,and H.Wang, "Trace norm regularized matrix factorization for service recommendation," in International Conference on Web Services,ICWS'13,pp.34-41,2013.

All Rights Reserved © 2016 IJARMATE