Atomic Web Services Reliability Technique For Service Oriented Architecture

Diksha M. Bhalerao [1], Prof.Dr Dayanand R. Ingle [2]
[1]Dept of Computer Engg.
[2]Professor and Head, Dept of Computer Engg
[1, 2] Bharati Vidyapeeth College of Engineering, University of Mumbai

Abstract- *To address the challenges in prediction of atomic web service reliability for the Service Oriented Architecture (SOA), in this paper we proposed the clustering based approach called the Dynamic Clustering (DCLUS). The DCLUS is based on recent work reported called CLUS. The novelty in DCLUS compared to CLUS technique is use of dynamic width clustering technique. In CLUS, k-means clustering method exploited for the users and services clustering. However, due to the limitations of k-means, the dynamic width clustering is proposed to optimize the performance of clustering and hence the prediction accuracy. The proposed DCLUS model for the reliability prediction of atomic web services that estimates the reliability for an ongoing service invocation based on the data assembled from previous invocations. With the aim to improve the accuracy of the current state-of-the-art prediction models, we incorporate user–, service– and environment–specific parameters of the invocation context.*

Keywords- K-means clustering,QoS Prediction, Reliability, Web Services

I. INTRODUCTION

Web service causes end clients to communicate with one another; additionally, it goes about as a stage for creating interoperable circulated applications, which permits software engineers to speak with other data suppliers, without making a big deal about backend or front-end errands. Execution of the utilized Web services enormously influences the execution of the service-situated systems. To speak with one another and customers web services are utilized, they additionally grant various applications to convey from different assets as well. They are exceptionally adaptable as they are not constrained to any operating system or programming dialects.Root Mean Square Error (RMSE) is the standard deviation of the prediction errors for commonly used in forecasting, climatology, regression analysis to check experimental results.Regularly these services are offered by outsider suppliers from different associations or then again endeavours that query service usage, providing service depictions likewise giving related specialized and business support [1]. Diverse services are created utilizing unmistakable advances that are conveyed over distinctive stages and are conveyed through different correspondence joins. Be that as it may, the Quality of Service (QoS) they offer may change despite the fact that their functionalities are comparative. A vital perspective towards worthiness of a web service is the means by which they meet the execution pre-requisite. As web service is one of the fundamental supporting hidden advancements in Service Situated Engineering (SOA), its execution should be examined principally. The conduct of Web Services is dynamic so that anticipating its response time amid beginning times of the Software Development Life Cycle (SDLC) turns out to be progressively unpredictable.

Instead of functional requirements, non-functional requirements, for example, QoS properties response time, throughput, reliability, disappointment rate and so on assume a crucial job in a client's necessity. QoS properties are dynamic in nature that changes often continuously. Thus, in service figuring, numerous kinds of research are presently day's completed on QoS forecast [2]. To build a dependable framework the engineer ought to guarantee the reliability of every person nuclear web service included. Change in QoS of nuclear web services may prompt a change in QoS of composite web service; it might debase the execution of a service-situated framework.

Among the previously mentioned QoS characteristics, for example, availability, reliability, and throughput and so on reliability is fundamentally picked as the forecast article because of its noteworthiness. It is on the grounds that they have a cozy relationship with equipment and programming arrangement, network conduct, load, client/service area which all by implication lead to the adjustment in the watched reliability esteem. Conduct of programming framework winds up strange when reliability neglects to meet essential necessity, it prompts produce immense a misfortune in a few spaces, for example, bank, military, aviation and so on that put an extreme interest on reliability. Client saw reliability and service determined reliability shift in like manner [3], [4]. This work centreson the reliability of nuclear web service, one of the fundamental non-functional properties. Service reliability can be characterized as the likelihood of effectively finishinga service summon under determined time requirements and conditions. It very well may be resolved to utilize past conjuring data tests as the proportion of various fruitful summons to the entirety of summons performed. A productive method to use these examples is to gather incomplete yet applicable example data from past summons and after that applying expectation algorithm for absent or unknown records. These examples are assembled by means of synergistic feedback or service observing [5]. Here a model named CLUS (CLU String) in view of the communitarian technique is presented; too the proposed clustering algorithm in CLUS display is supplanted by a propelled K-Means algorithm which is exceptionally useful for high dimensional data [6].

The paper has been divided as follows: section II explains the Literature Survey, section III deals with related work, section IV methodology, section V describes results and descriptions followed by conclusion and then References.

.

II. LITERATURE SURVEY

The Service Oriented Architecture (SOA) enables web application designed to compose the atomic services into more complex ones in order to deliver more advanced functionalities. While constructing composite services, it is essential for the developer to select high quality atomic service candidates. The application quality relies on both functional and non-functional qualities of the selected candidates. Hence, to create an efficient composite application, the developer should be provided with reliable information on both atomic services' functionalities and their non-functional dependability attributes, such as response time, reliability and availability. The atomic web services reliability is one of the most challenging tasks while constructing QoS-aware composite work-flows based on SOAs. This problem can be solved by designing the effective solution for reliability prediction of atomic web services that estimates the reliability for an ongoing service invocation based on the data assembled from previous invocations.

The aim of this research work is to proposed novel prediction method for Atomic web service reliability model. The objectives are:

- To discuss the importance of analyzing the web service reliability
- To study and analyze the various techniques for web service reliability prediction
- To design new efficient and scalable algorithms for atomic web service reliability for the recommendation systems.
- To design, implement, and evaluate the proposed technique.

III. RELATED WORKS

To evaluate web service reliability qualities picked are distinctive for various prediction models or strategies. Here, different web service reliability prediction methodologies dependent on different parameters are examined. Web services are dynamic in nature that makes them functionalities accessible over various interfaces over the web. Numerous scientists more often than not concentrate on examining service reliability while planning new models for the service-arranged framework [7]. Studies uncover that gathering test data for this task is troublesome. As the quantity of parameters utilized for producing test data expands, it in swings builds the prediction exactness. From many existing ways to deal with reliability prediction, community-oriented separating is one of the productive methods [8]. Communitarian separating is fundamentally ordered into memory-based, demonstrate based and crossover. Existing works demonstrate that community-oriented sifting approaches result in a promising outcome [9]. Be that as it may, their principle drawback of is its adaptability and exactness issue, likewise, they need extra putting away space for each watched service-client esteem pair. Such a methodology does not scale when a huge number of clients and service happen. As expressed above prediction exactness relies upon an assortment of elements, for example, they may have an effect when they are considered or not.

2.1 Affecting elements

As the web will dependably vary contingent upon different ecological variables or equipment assets, results of service summons likewise rely upon such factors. Because of this dynamic nature of web service, the greater part of the existing customary strategies is not reasonable for deciding the reliability of web service conjuring. While building up a model for reliability prediction different methodologies pick different parameters or variables, they produce a distinctive effect on the diverse model.

Nonfunctional nature of service is principally impacted by the area of service and the client's [10]. Another factor that impacts the framework execution is the inward service multifaceted nature; conceivably it impacts the service unpredictability additionally [11]. Time of conjuring of service has hugeness as the heap in multi-day shifts as needs are [12].

2.2 Diverse techniques for web service QoS prediction

Composite service is made out of a lot of atomic services. As we know, regardless of whether distinctive web services have comparable functionalities, their nonfunctional properties (QoS) may fluctuate. For an appropriate choice of significant atomic web services for a composite service its QoS esteem should be anticipated, on the off chance that the estimation of QoS is more prominent, at that point the service will be solid [13].

Another well-known technique dependent on community-oriented sifting is Matrix Factorization (MF) which is as of late utilized for web service reliability prediction likewise [14]. It makes utilization of the client's area and network map for prediction. Network map is utilized for estimating the separation between clients in the network. Missing QoS qualities can be anticipated by building a gathering of non-negative inactive factor (NLF) models, it produces unknown QoS data esteem with high precision from the recently invoked history of web services [15].

As a rule, QoS estimations of unknown services are anticipated by utilizing the known QoS benefits of existing clients. Sometimes they may prompt mistaken outcomes if the QoS esteems are taken from questionable clients [16]. To address this issue, the notoriety of QoS esteem contributing clients is first determined and is then utilized in the Notoriety based Matrix Factorization (RMF) [17].

To foresee the availability of atomic web services, a model named as LUCAS (service Burden, Client area, service Class, Service area) is presented [18]. The procedure begins with the characterization of the gathered past summons data, in view of the clients' and services' geographic areas, the heap of the service supplier at the real time of the conjuring, and the computational requirements of the invoked service. Demand order is done dependent on recently characterized gatherings when another progressing service asks for is gotten. Closeness proportion of that summon with other recently watched summons is done, which along these lines decides the most comparable arrangement of substances. In light of the outcomes acquired, the assessed availability of a service conjuring is determined by considering the effects of the four LCS parameters.

A reliability prediction display named CLUS has utilized to gauges the reliability for a progressing service conjuring dependent on the data collected from past summons [19]. The reliability prediction process is done in two stages: a data clustering stage and a prediction stage. Clustering of the history summon test is performed before prediction, K-Means clustering algorithm is utilized for the procedure. In light of the natural conditions, the time windows in multi-day are clustered dependent on the reliability execution from the past summon tests. Thus, clients and services are additionally clustered considering their reliability execution inside each time window cluster. At last, a three-dimensional space D containing clustered data is made. In the wake of finishing the clustering stage, prediction of the atomic services reliability can be performed in the prediction stage utilizing a straight relapse algorithm. To lessen the versatility issues present in the cutting edge approaches, the past conjuring data are collected utilizing a K-means clustering algorithm. This model delivers increasingly versatile and exact predictions contrasted with already existing techniques.

CLUS demonstrate addresses different impediments in community sifting based methodologies, for example, exactness and adaptability issues. LUCAS demonstrates is progressively relevant just when the info parameters are exceedingly accessible. Though, so as to expand the exactness of CLUS prediction demonstrate, the K-Means algorithm utilized in CLUS is supplanted by our proposed work.

IV. METHODOLOGY

Here a review of CLUS with dynamic clustering model for atomic web services reliability prediction is presented. To make a prediction model progressively exact, required parameters ought to be chosen shrewdly. In the CLUS model, user, service, and environment clustering are picked which makes a difference to correctly decide service summon setting more compelling than other prediction models. To amend the versatility issue in a current model the gathered summon tests are assembled into three unique measurements related with three unique parameters (User, Service,Environment) utilizing the dynamic clustering approach. A dynamic clustering approach is developed from K-means calculation's thought. A diagram of the prediction model is shown in Figure the flowchart for the proposed Atomic web service reliability framework using dynamic clustering approach. This expects to create higher precision and effectiveness of the prediction model.

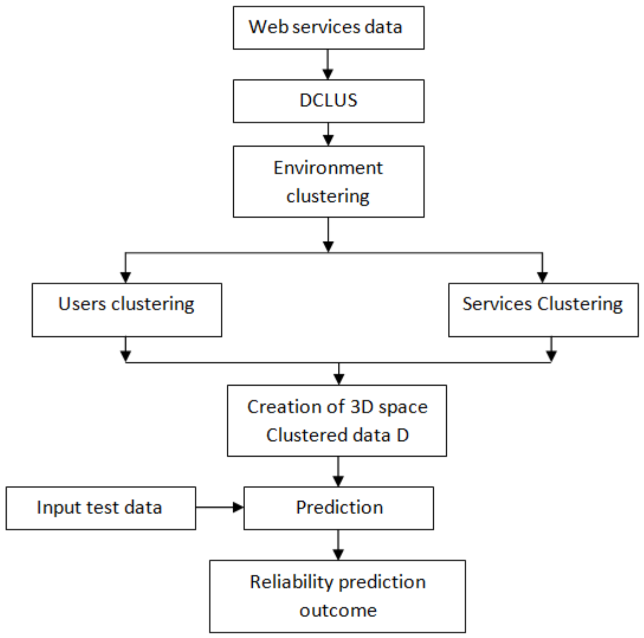

Figure: 1 shows the flowchart for the proposed Atomic web service reliability framework using dynamic clustering approach.

To make reliability predictions more scalable and accurate for future invocations data needs to be transformed into a more structured and compact form. Steps in data clustering process of CLUS defines how prediction is performed from clustered data. A service invocation can be defined as R (u,s,t). u is the user who is executing the service, s is the invoked service and t is the time of invocation. Past invocation sample contains the above-addressed parameters. For that data are stored into a three-dimensional space D [u,s,e] were u, s,and e each group of parameters. Based on these clustereddata further prediction is performed.

The proposed reliable data prediction perform in two phases first phase data collection& clustering phase is shown algorithm 1 below and a second phase is prediction phase and data clustering phase modified shown in algorithm 2.

Phase 1
Proposed Algorithm 1:Dynamic Clustering using K-Means.
(Data Collection and Data Clustering Phase)
Input: Take the web services data.
Result: Clustered Data

A. Step 1: Collect the web services from different cloud platforms with their all properties.
B. Step 2: Apply the environment specific parameters clustering using k-means.
C. Step 3: Apply the user specific parameters clustering using k-means.
D. Step 4: Apply the service specific parameters clustering using k-means.
E. Step 5: Display the outcomes of all clustering steps

Phase 2
Proposed Algorithm 2: Predication Phase
Input: Clustered Data
Result: Create 3-D space matrix,Predict the data and measure RMSE and prediction time.

A. Create 3-D space matrix D using E, U, and S data
B. Design prediction model
C. Test data preparation and prediction
D. Measure the RMSE and prediction time

4.1 User Clustering

In User clustering, a few user bunches need to be characterized. User-explicit properties incorporate different factors, for example, the user's area; organize use, gadget abilities, and user profiles that may affect the unwavering quality of administration. In a request to consolidate user-explicit parameters into the prediction show, users are grouped dependent on the dependability execution dependent on dynamic K-Means Clustering Algorithm.

User data clustering, every user assembles uk contains users having reliability execution. Every individual user contains an n- dimensional response time and throughput should be determined. In the wake of ascertaining RMSE esteem and appointing them to singular users, dynamic clustering is performed. Users into various users bunch as indicated by the similitude in RMSE esteem. This serves to effortlessly correspond existing conjuring tests to a suitable user assemble for a forecast.

4.2 Service Clustering

Service-explicit parameters speak to the effect of service qualities on reliability performance. Factors, for example, service's location, service's computational multifaceted nature, and framework assets, for example, CPU, RAM, disk and I/O tasks may incorporate. Here, just the service's location is considered as service-explicit parameters for the forecast process. At long last, services are clustered dependent on the reliability performanceon dynamic K-Means Clustering Algorithm.

Service clustering process is like that of user clustering. Here, a few servises bunches sj need to characterize which contain administrations having comparable or same reliability execution. Every individual servies s containsan n-dimensional response time and throughput should be determined. It contains the normal reliability of administration s conjured amid an environmental condition ei. In the wake of deciding the n-dimensional grid, administrations are bunched into a few administration bunches utilizing dynamic clustering dependent on their RMSE esteems. Presently each accessible past summon records can be effectively associated to the suitable administration gathering.

4.3 Environment Clustering

It indicates certain environment-explicit parameters identified with the current environmental conditions, for example, arrange execution, specialist organization load at the season of a conjuring. Because of down to earth restrictions, administration load is as it was considered as environment parameter. Administration burden can be characterized as the number of solicitations in a second. User observed values for QoS properties shift broadly for various clients affected by heterogeneous client environments or capricious Web associations. Changes in-administration load fundamentally impacts QoS factors, for example, accessibility or then again unwavering quality. Throughout the multi-day, significant burden variety may happen, so the day is separated into a self-assertive number of time windows accepting the heap to be consistent for each time window and the past conjured tests are scattered over them [20]. It makes strides prediction precision.

The dataset contains a few normal dependability esteems in view of all the three unique parameters referenced previously. For the prediction procedure, we have to group the accessible dataset for each particular parameter. In environment, explicit information grouping n number of particular environmental conditions (E) is determined dependent on various burdens, E = {e1, e2,...,en}, where e indicates the environmental condition dependent on the specialist organization load. After deciding the time window as expressed above, normal dependability 1 esteem for each time wi should be determined.

$$pw = \frac{1}{|Wi|}\sum pr$$

Where Wi is the set of records for the time, r is supplication sample. Pr is user perceived reliability for that supplication. When k is the number of environmental conditions exists, applying the K-Means algorithm, we required to partition the data points into K-different clusters cluster representing each environmental condition.

4.3 Creation of 3D space Clustered data D

In 3D space, After completion of clustering phase each record r(u,s,t) be associated ui,si,ei with the corresponding data clusters. Then space D can be generated by:

$$D[ui,si,ei] = \frac{1}{|R|}\sum_{r\epsilon R} pr$$

Here pr represents the user perceived reliability for an invocation r and R is set,

$$R = \{r(u,s,t)|r \in ui \cap r \in si \cap r \in ei$$

Assume, on the off chance that we need to anticipate the normal unwavering quality p1 of a progressing web administration conjuring r1(u1, s1, t1). Among all the environmental conditions groups produced in environment explicit bunching process, the normal unwavering quality of all nearest environment conditions group is determined. It maps them to the comparing load condition in the environment. When they are related with the real burdenenvironment conditions bunch w1, we have to check whether there is a set S in the past conjuring test which contains records with a similar summon setting parameters of progressing administration r1.

$$S = \{rs|us = us \cap ss = ss \cap ts \in w1\}$$

If S is non empty the reliability value $p_s$is measured using the existing reliability values in the set S

$$ps = \frac{1}{|S|}\sum_{r\epsilon s} pr$$

So, whenever set S is unfilled, figure the unwavering quality pc utilizing the information put away in the space D as, p1 = D[u1,s1,e1].Space D should be updated when each time. In this way RMSE reliable value is estimated for prediction.

V. RESULTS AND DISCUSSION

The implementation of the suggest K-Means clustering algorithm is assessed and separates at negating availableun require data.

At this experiment carried out to compare the CLUS model using K-Means and CLUS model using dynamic clustering approach. This research based on real data using different services. We take several users geographical locations placing 100 web services.

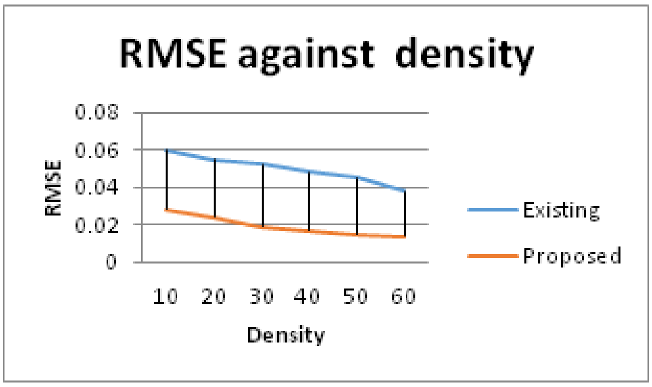

Figure 2: RMSE against density

For calculating average RMSE values for existing system is 0.052 and average RMSE values proposed system is 0.019. The lowest RMSE value is better to performance of system.

Table 1: Compare Result of RMSE values.

No of Iterations	Existing System RMSE values	Proposed System RMSE values
1	0.070	0.028
2	0.050	0.024
3	0.062	0.019
4	0.048	0.017
5	0.050	0.013
6	0.037	0.015

VI. CONCLUSION

As web services are arrange self-ruling and comprehensively used in various extremely incredible in degree purposes, the amount of web service users is growing rapidly. So that, to make a brilliant service-arranged structure, foreseeing its reliability is basic. It is a pervasive QoS factor. These work examinations distinctive QoS regard desire procedures of web services for studying its quality and to ensure its reliability. Our model checks the reliability for a nonstop service conjuring in light of the data assembled from past request by in-planning user, service, and environment–express parameters of the conjuring setting. A mixture of the past conjuring data is done using dynamic clustering approach which is strong for gathering high dimensional data. The evaluation results attest that CLUS using dynamic clustering model makes increasingly exact desires when appeared differently in relation to using K-Means estimation. It is assessed by processing Root Mean Square Error (RMSE).

REFERENCES

[1] T. Yu, Y. Zhang, and K.-J. Lin, “Efficient algorithms for web services selection with end-to-end QoS constraints,” ACM Trans. Web, 2007.

[2] M. P. Papazoglou, “Service-oriented computing: Concepts, characteristics and directions,” in Web Information Systems Engineering, 2003. WISE 2003. Proceedings of the Fourth International Conference on. IEEE, 2003, pp.3–12.

[3] Z. Zheng, H. Ma, M. R. Lyu, and I. King, “Qos-aware web service recommendation by collaborative filtering,” IEEE Transactions on Services Computing,2011.

[4] Hongbing Wang, Haixia Sun and Qi Yu, “Reliable Service Composition via Automatic QoS Prediction”, IEEE2013.

[5] X.Luo,Y.-N.Xia,andQ.-S. Zhu, “Incremental collaborative filtering recommender based on regularized matrix factorization,” Knowl. Based Syst., vol. 27, pp. 271–280,2012.

[6] Marin Silic, Goran Delac, and Sinisa Srbljic, “Prediction of Atomic Web Services Reliability for QoS-aware Recommendation”, IEEE 2014.

[7] X. Su and T. M. Khoshgoftaar, “A survey of collaborative filtering techniques,” Adv. in Artif. Intell., vol.2009.

[8] L. Baresi and S. Guinea, “Event-based multi-level service monitoring,” in ICWS, pp. 83–90,2013.

[9] Klein, F. Ishikawa, and S. Honiden, ‘‘Towards NetworkAware Service Composition in the Cloud,’’ inProc. 21st Int’l Conf. World Wide Web, 2012, pp.959/968.

[10] N.B. Mabrouk, S. Beauche, E. Kuznetsova, N. Georgantas, and V.Issarny,‘‘QoS-AwareServiceCompositioninDynamicService Oriented Environments,’’ in Proc. 10th ACM/IFIP/USENIX Int’l Conf. Middleware, 2009, pp.123/142.

[11] Y. Wang, W.M. Lively, and D.B. Simmons, ‘‘Web Software Traffic Characteristics and Failure Prediction Model Selection,’’ J. Comput. Methods Sci. Eng., vol. 9, no. 1, pp. 23-33, Apr2009.

[12] “Service Selection for Web Services with Probabilistic QoS” , IEEE transactions on services computing, vol. 8, no. 3, may/june2015.

[13] Mingdong Tang et al., “Collaborative Web Service Quality Prediction via Exploiting Matrix Factorization and Network Map”, IEEE2015.

[14] Li, X. Fan, Y. Zhou, and Z. Su, “Evaluating the reliability of web services based on bpel code structure analysis and runtime information capture,” in Asia Pacific Software Engineering Conference 2010, IEEE Computer Society,2010.

[15] Xin Luo et al., “Generating Highly Accurate Predictions for Missing QoS Data via Aggregating Nonnegative Latent Factor Models”, IEEE 2015.

[16] Y. Xu, J. Yin, and W. Lo, “A unified framework of QoS-based web service recommendation with neighborhood-extended matrix factorization,” in Proc. 6th IEEE Int. Conf. Service Oriented Computing and Applications (SOCA 2013), 2013, pp.198–205.

[17] L. Zeng, B. Benatallah, A. Ngu, M. Dumas, J. Kalagnanam, and H. Chang, “Qos-aware middleware for webservices composition,” Software Engineering, IEEE Transactions on,2004.

[18] Jianlong Xu, Zibin Zhen, and Michael R. Lyu, “Web Service Personalized Quality of Service Prediction via Reputation-Based Matrix Factorization”, IEEE transactions on reliability, vol. 65, no.1, march2016.

[19] Xin Luo et al., “Generating Highly Accurate Predictions for Missing QoS Data via Aggregating Nonnegative Latent Factor Models”, IEEE 2015.

[20] Marin Silic, Goran Delac, and Sinisa Srbljic, “Prediction of Atomic Web Services Reliability for QoS-aware Recommendation”, IEEE, 2015.