Advances in Computer Science Research, volume 65

3rd International Conference on Communications, Information Management and Network Security (CIMNS 2018)

Exploration and Research on Prediction Methods of Web Service QoS

Chengcheng Lou and Yi Sun*

School of Software Engineering, State Key Laboratory of Networking and Switching Technology Beijing University of Posts and Telecommunications

*Corresponding author

***Abstract*—The web service QoS (quality of service) plays an important role in the evaluation and selection of web services. [1,2,3,4] In previous research on QoS prediction, there is a lack of a model that can dynamically predict the state of web services, and there is no revision of the prediction results. In this paper, an exponential smoothing based model for state prediction of web services is established, and a prediction correction method based on user evaluation is proposed. Combining the prediction results and correction methods, the accurate prediction of web service QoS is finally realized. Experimental results show that this scheme can effectively improve the prediction accuracy of web service QoS.**

Keywords—Web Service Quality; exponential smoothing; user rating; prediction

I. INTRODUCTION

This Web Service QoS is a comprehensive evaluation standard of web Service quality. In order to provide users with better Service experience, suppliers need to dynamically predict the value of web Service QoS. This paper proposes a prediction method based on the state of web services and user evaluation. This method uses the exponential smoothing method to predict the busy index of web services and calculate the similarity between the target service and other services. According to the similarity, preliminary QoS prediction results can be obtained. Then, the similarity of different users is calculated according to the users' QoS evaluation of the service, and the QoS predicted value is corrected by using similar users' evaluation of the target service to further improve the prediction accuracy.

II. DEFINE THE WEB SERVICE STATUS

A. The Definition of Web Service State as Follows

The state of a web service depends on several factors. Among them, the busy degree of web service depends on the load of server and the maximum concurrent visits of service. [5] Server load reflects the use of hardware resources, mainly related to CPU utilization, disk IO occupancy, and server memory. The maximum concurrent visits to a web service are limited by server conditions and the environment in which the service occurs. [6, 7] The network environment of the server can also affect the quality of web services, and the QoS values obtained by running the same service on different servers can vary greatly. In addition, the task type and task size on the web service also determine the similarity between the two services. The description of web services is given based on the above analysis.

$$S=\{L, CV, la, lo, TT, TS\}$$

Where S represents a collection of service states. L represents server load.CV stands for concurrent access to services. La represents the latitude to run the server. Lo represents the longitude of the running server.TT represents the task type. TS denotes the task size.

B. Calculate the Busy Index and Relative Distance

By definition, the busy index of a web service is calculated by load and concurrent visits:

$$b = \begin{bmatrix} L \\ cv \end{bmatrix}^T * \begin{bmatrix} w1 \\ w2 \end{bmatrix}. \quad (1)$$

Where b is the busy index of web services. W1 and w2 represent weight coefficients of load and weight coefficients of concurrent hit rates.

The physical location of the server is determined by latitude and longitude, and the distance between the two servers is calculated by the following formula [8].

$$D_{0j} = 111.99 * \sqrt{\left(p_{0_{la}} - p_{j_{la}}\right)^2 + d}. \quad (2)$$

$$d = \left(p_{0_{lo}} - p_{j_{lo}}\right)^2 * \cos^2\left(\frac{p_{0la} + p_{j_{la}}}{2}\right). \quad (3)$$

The geographic location of the Web service p = {la, lo} la: stands for dimension lo: stands for longitude. The relative distance between the two services is calculated according to the location information P0 and Pj.

When comparing the similarity of the two services, firstly, they are divided according to task type and task quantity, and then the busy index and relative distance are obtained by using formulas (1) and (2), and then the similarity is calculated.

III. EXPONENTIAL SMOOTHING FORECASTING MODEL

A busy index sequence for the target service can be obtained from the server history. In order to apply the exponential smoothing method, the non-stationary sequence is transformed into a stationary one by difference [9].

Set the time series to X(t) (t=1,2,…,n), X(i), X(i+1)are time series variables that are spaced one step apart, and the first difference of x (i + 1) is:

$$\Delta X(i+1) = X(i+1) - X(i). \quad (4)$$

The d-order difference formula can be deduced based on first-order difference:

$$\Delta^{d} x(i+1) = \Delta^{d-1} x(i+1) - \Delta^{d-1} x(i). \quad (5)$$

Usually the result of two differences is a stationary time series. The static time series can be divided into three types: 1. There is no obvious trend change; 2. Linear trend; 3. Seasonal trends.

For case 1, use the one-time exponential smoothing method to predict:

$$b_t = \lambda a_{t-1} + (1-\lambda) b_{t-1}. \quad (6)$$

Where b_t is the smooth value of time t. This is the smoothing constant. a_t is the actual value of time t.

For case 2, the second smoothing value can be obtained by smoothing again on the basis of the first exponential smoothing. The formula is as follows:

$$b_{t+m} = \left(2 + \frac{\lambda m}{1-\lambda}\right) b_t - \left(1 + \frac{\lambda m}{1-\lambda}\right) a_t$$

$$= 2b_t - a_t + \frac{m\lambda(b_t - a_t)}{1-\lambda}. \quad (7)$$

Where m is the predicted step size. $b_{(t+m)}$ is the prediction of time (t+m).

For case 3, the tri-exponential smoothing method can be obtained by deriving formulas (6) and (7).

$$b_{t+m} = a + b + c. \quad (8)$$

$$a = 3b_t^1 - 3b_t^2 + b_t^3. \quad (9)$$

$$b = \frac{[(6-5\lambda)b_t^1 - (10-8\lambda)b_t^2 + (4-3\lambda)b_t^3]\lambda m}{2(1-\lambda)^2}. \quad (10)$$

$$c = \frac{(b_t^1 - 2b_t^2 + b_t^3)\lambda^2 m^2}{2(1-\lambda)^2}. \quad (11)$$

$$\begin{cases} b_t^1 = \lambda a_t + (1-\lambda) b_{t-1}^1 \\ b_t^2 = \lambda b_t^1 + (1-\lambda) b_{t-1}^2 \\ b_t^3 = \lambda b_t^2 + (1-\lambda) b_{t-1}^3 \end{cases}. \quad (12)$$

Where b_t^1 is the first exponential smoothing, b_t^2 is the second exponential smoothing and b_t^3 is the third exponential smoothing.

In order to implement the model, it is necessary to select appropriate parameters and smooth values of the first cycle. Since the weight of the early data is very small, the selection of the first-stage smoothing values has little impact on the predicted results. In general, you can use the mean of the previous period data as a phase smoothing value.

$$b_1 = \frac{\sum_{i=1}^{m} a_i}{m}. \quad (13)$$

Where a_i represents the actual value of I for the period.

The predicted value comes from the weighted sum of the previous actual value, and the most recent data has more weight. The accuracy of prediction can be improved by selecting the appropriate parameter λ. We divide [0,1] equidistant into 100 parts, and the results are represented by vectors ε. When the parameter λ is equal to the ith value of vector ε, the mean square error of predicted value and actual value is represented by $\hat{y}(\varepsilon_i)$.

$$\hat{y}(\varepsilon_i) = \frac{1}{N}\sum_{t=1}^{N}(a_t - b_t)^2. \quad (14)$$

Where N represents the sequence number of time series. The formula (14) is used to calculate the error corresponding to different smoothing constants, and select the smoothed constant with the minimum error as the optimal solution.

IV. Predicted QoS Value Of The Target Service

The first step is to calculate the similarity between the target service and other services. The target service is set to g = {L, CV, la, lo, TT, TS}. The reference service is set to s ={L, CV, la, lo, TT, TS}. Select a set of services that have the same task type as the target service, represented by sj(j=1,2,…,m). Calculate the similarity between service sj and g.

$$sims(g, sj) = \begin{bmatrix} si1(g_b, sj_b) \\ si2(D(g, sj)) \\ si3(g_{TS}, sj_{TS}) \end{bmatrix}^T * \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix}. \quad (15)$$

$$si1(b_0, b_1) = 1 - \frac{|b_0 - b_1|}{\max(b)}. \quad (16)$$

$$si2(d_{g,sj}) = \frac{d_{g,sj}}{sum(d)}. \quad (17)$$

$$si3(s1, s2) = 1 - \frac{|s1 - s3|}{\max(s)}. \quad (18)$$

Where g_b represents the busy index of the target service, which can be obtained from the web service busy index prediction model. sj_b represents the busy index of the service sj retrieved from historical data. D (g, sj) represents the distance between services, which can be calculated by formula (2). g_{TS}, sj_{TS} represents the number of tasks. Max (b) represents the maximum value of a busy index. Sum (d) is the Sum of the relative distances. Max (s) represents the maximum number of tasks in all services. $\begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix}$ is the weight vector.

Then, we select the top K services based on their similarity and use them to calculate the QoS predictions for the target service.

$$QoS_{s0} = \sum_{i=1}^{k} \frac{\delta + sims(g, si)}{\sum_{j=1}^{k}(\delta + sims(g, sj))} Q_0 S_i. \quad (19)$$

Where QoS_{s0} represents the predicted QoS value of the target service and QoS_i represents the latest QoS record value of the ith service in top-k service.

V. Revise Prediction Results Based On User Evaluation

In the traditional similarity calculation method based on user evaluation, the user's evaluation of the service is only taken as the measurement standard, ignoring the impact of the service itself on the evaluation. When most users rate a service highly, new users tend to rate the service blindly. In order to compare the similarities between two users, the impact of the service itself must be reduced for evaluation. The global cumulative evaluation index of the service is introduced to reflect the overall evaluation of the service by all users. All service sets S = {s0, s1,…, sn}, all user sets U = {u0, u1, ..., um}. Vector QSi = $[qs_0, qs_1, \ldots, qs_n]^T$ is used to record user I's evaluation of all services. Services that are not evaluated by the user will be populated with the average of other evaluation services. $QS1_j$ represents user 1's QoS evaluation of service j(j=1,2,…,n). The global cumulative assessment index for service j is shown below.

$$\text{sigmoid}_j = \frac{\sum_{i=1}^{m} QSi_j}{m}. \quad (20)$$

The user similarity was calculated based on the Pearson product-moment correlation coefficient,

$$\text{simu}(u_0, u_i) = \frac{\sum_{j=0}^{n} \frac{1}{sigmoid_j}(QS0_j - \overline{u}_0) * (QSi_j - \overline{u}_i)}{\sqrt{\sum_{j=0}^{n}(QS0_j - \overline{u}_0)^2} * \sqrt{\sum_{j=0}^{n}(QSi_j - \overline{u}_i)^2}}. \quad (21)$$

Where u_0 , u_i represent the target user and other users. $\overline{u}_0$ represents the average evaluation of all services by target user. $\overline{u}_j$ represents the average evaluation of all services by user j. Select the top k users according to the similarity.

QoS prediction was calculated based on similarity and historical assessment.

$$\text{QoSu} = \overline{u}_0 + \frac{\sum_{i=1}^{k} simu(u_0, u_i) * (QSi_s - \overline{s})}{\sum_{i=1}^{k} simu(u_i, u)}. \quad (22)$$

Where $\overline{s}$ represents the average evaluation of service s by top-k users.

The predicted results are revised according to formulas (22) and (19):

$$\text{QoS} = \begin{bmatrix} \text{QoS}_{s0} \\ QoSu \end{bmatrix}^T * \begin{bmatrix} \beta_1 \\ \beta_2 \end{bmatrix}. \quad (23)$$

Where $\begin{bmatrix} \beta_1 \\ \beta_2 \end{bmatrix}$ represents the weight vector.

VI. Application

To verify the effectiveness of this method, a simulation experiment based on campus Web service platform is designed. In this experiment, 20 sets of Web services were selected, each consisting of Web services of the same task type. Record information about these services, including number of tasks, number of concurrent visits, geographic location, server load, and response time of user call services. The load is represented by server CPU utilization. Firstly, the prediction model of Web service busy index is established, and then the prediction model is used to predict the future Web service busy index. Finally, the QoS value of the service is predicted based on the service status. The following formula is used to calculate the prediction accuracy:

$$\text{RTP} = 1 - \frac{|R_p - R_a|}{R_a}. \quad (24)$$

R_p and R_a represent the predicted value and the actual value respectively.

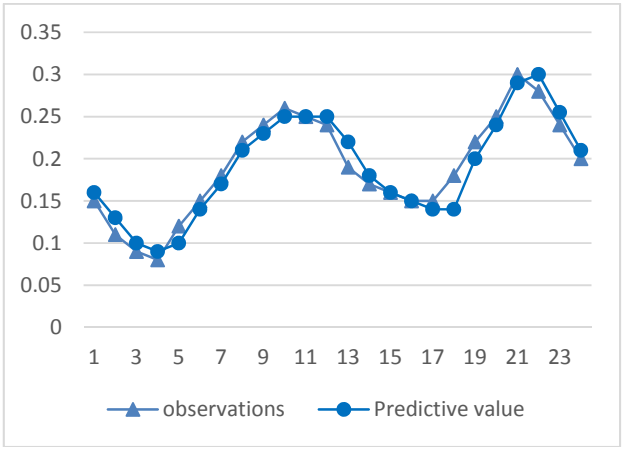

FIGURE I. PREDICTION OF WEB SERVICE BUSY INDEX WITHIN 24 HOURS.

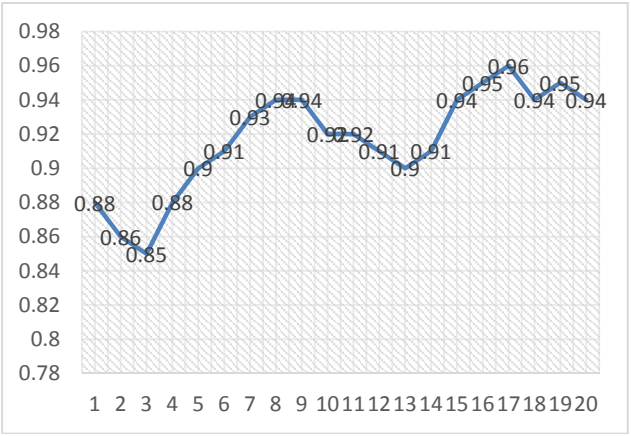

FIGURE II. PREDICTIVE ACCURACY OF WEB SERVICE RESPONSE TIME.

VII. Conclusion

Traditional Web Service QoS prediction methods cannot dynamically consider the state of Web services. Based on the exponential smoothing method, a dynamic prediction method that can use historical information comprehensively is proposed, and the shortcomings of traditional similarity calculation method based on user evaluation are pointed out. In the future research,

factors affecting the state of Web services can be further explored to improve the accuracy of Web services QoS prediction

ACKNOWLEDGMENT

This research was financially supported by the 2018 Double-class notch talent competition Project.

REFERENCES

[1] Zhizhong Liu, Zhijian Wang, Xiaofeng Zhou, Yuansheng Lou, Ling Shang. Study on dynamic prediction of Web service QoS based on case reasoning [J]. Computer science,2011,38(02):119-121+137.

[2] Wei Chen. Prediction of Web service QoS based on collaborative filtering [J]. Technology innovation and application,2014,(05):69.ISSN: 2095-2945

[3] Yajun Leng, Qing Lu, Changyong Liang. Overview of collaborative filtering recommendation technology [J]. Pattern recognition and artificial intelligence,2014,27(08):720-734.ISSN: 1003-6059

[4] You Ma, Shangguang Wang, Qibo Sun, Fangchun Yang. A Web service QoS measurement algorithm that comprehensively considers subjective and objective weights [J]. Software journal,2014,25(11):2473-2485.

[5] Hong Zhao, Jian Lin, Miaoliang Zhu. Dynamic load balancing model for Web services [J]. Computer engineering and design,2006,(21):4108-4110.

[6] Xiangyun Lin, Xiaoqing Liu, Mingdong Tang, Buqing Cao, Jianxun Liu. Empirical research on the correlation between Web service QoS and user location [J]. Computer engineering and science,2013,35(09):83-88. ISSN:1007-130X

[7] Yan Hai, Zhijian Wang, Zhizhong Liu, Xiaofeng Zhou, Ling Shang. A method supporting dynamic prediction of Web services QoS [J]. Journal of Nanjing university of science and technology,2013,37(01):52-59.

[8] Lienert B R, Havskov J. A computer program for locating earthquakes both locally and globally[J]. Seismological Research Letters, 1995, 66(5): 26-36

[9] James D.Hamilton., Time Series Analysis, Xiaohua Xia,Beijing,2015,49-68