

Research on the Improvement of Traditional Linear Weighted Algorithm for QoS-based Web Service Selection

Yuqiang Li¹, Wei Zhao^{1*} and Lei Che²

¹*School of Computer Science and Technology, Wuhan University of Technology, Wuhan 430063, China*

²*School of Information Management, Beijing Information Science & Technology University, Beijing 100085, China*

{liyqiang, maszhaowei}@whut.edu.cn, chelei@bistu.edu.cn

Abstract

At present, quality of service (QoS) has become an important guidance in selecting a satisfactory web service when many similar web services provide overlapping or identical functionalities. Therefore people have carried out many studies on definition and standardization of QoS properties and QoS-based selecting algorithms of web service. Current web service selection usually adopt linear weighted algorithm. But it still has some limitations. This paper proposes a new QoS-based algorithm developed from traditional linear weighted algorithm for web service selection. The main improvements include removing distinction of QoS properties value tendency, selecting fixed value and a hybrid subjective-objective approach to set the weight of each QoS property. We also present the analysis of the time complexity. Finally we do a comparison experiment between our algorithm and traditional algorithm. The results show that the improvements are feasible and our algorithm has better performance.

Keywords: *web service selection; QoS; linear weighted algorithm*

1. Introduction

More and more web services that provide overlapping or even identical functionalities are published throughout the internet nowadays. Hence, it's become very important for the service users to choose suitable web service satisfying their requirements from a set of candidate services. The methods of selecting a satisfactory web service are generally based on quality of service (QoS). The present researches on QoS include definition, calculation and standardization of QoS properties. Nevertheless, there are still many issues to deal with. For example, different tendencies of QoS properties value (some QoS properties have positive value tendency, *i.e.*, the higher the value, the higher the quality of service, while the other have negative value tendency, *i.e.*, the lower the value, the higher the quality of corresponding QoS-based algorithm to solve them. The remainder of this paper is organized as follows: Section 2 summarizes the related research efforts in this field. Section 3 introduces the definition and calculation methods of QoS properties. Section 4 makes a description of the standardization of the QoS properties of a web service. A calculation

* Corresponding author at: School of Computer Science and Technology, Wuhan University of Technology, Wuhan 430063, China.

Email address: maszhaowei@whut.edu.cn(W. Zhao).

method of QoS value of a web services is given in Section 5 and its algorithm complexity will be discussed in Section 6. In Section 7, we compare our algorithm with the traditional linear weighted algorithm and summarize the experiment results. Finally in Section 8, conclusions will be made along with prospective plans for our further study.

2. Related Work

As early as 2002, some researchers began to concentrate on the study of the definition and calculation of QoS properties. O'sullivan *et al.*, [1] were inspired by the phenomenon that customers use the information on the product label, together with the price, to make a rational purchasing decision and pointed out the necessity to add QoS attributes description to the description of web services. At the same time, they also stated that QoS is a comprehensive concept which includes a large number of non-functional attributes of web services, including execution price, availability, reliability, execution time, *etc.*, on the basis of the research result achieved by Chung [9]. Furthermore, they also made it clear that their non-functional attributes are of great importance to the discovery, invocation, substitution, combination, *etc.*, of services. However, people usually present different QoS model form, that is, different sets of QoS properties. For example, Sheth *et al.*, [2] present a four-dimensional QoS model consisting of time, cost, reliability and fidelity. The team led by Zeng also proposed their QoS model, including execution time, execution price, reputation, successful execution and availability in his representative article [3].

Researchers have conducted a lot of researches on definition, standardization and calculation about QoS properties of web services. And they have their own comprehension of the composition of QoS properties. In order to reach a consensus and unify related research work, W3C published a recommended standard of QoS properties requirements which underlies further related studies.

After reaching the consensus of QoS properties composition of web services, people begin to focus on the attributes of QoS properties. One of the aspects is how to calculate the QoS value in order to provide guidance for selecting web services. Menascé [7] studied a series of measurement issues on this aspect, such as “what to measure”, “how to measure”, “who to measure” and “where to measure” *etc.* Then he gave answers in Ref. [8] on some of above issues. And he also pointed out that QoS property is a composition of several quality attributes and its evaluation should be at the both sides of service providers and users. His research work indicated the direction for the subsequent related research.

The subsequent researchers conducted more researches. Yau *et al.*, [12] proposed their approach that selects the service that best satisfies users QoS requirements instead of the service with the best QoS which may be much overqualified for the users' QoS requirements. Researchers also use prospect theory to reflect the fluctuation of user feeling in order to determine how well a service satisfies a user's requirements [13].

3. Definition and Computing Methods of QoS Properties

Because of the existence of competition, it is inevitable that some of the web service providers provide overlapping or identical functionalities. So it is vital for users to choose a satisfactory web service from the candidate services. At present it is common to help carry out this process based on QoS. QoS is a comprehensive concept that includes lots of non-functional properties, such as execution price, availability, reliability, execution time, *etc.*, [1-4]. In order to implement web services selection on the basis of QoS, it is necessary to make the definition and calculation of QoS properties explicit and reach a consensus between service providers and service requesters. In order to simplifying the process of the algorithm,

we select four representative QoS properties of web service from the thirteen properties presented by W3C recommended standard [5]. We transform the properties having positive value tendency to negative one and keep the negative value tendency. Consequently, all these four QoS properties have the same value tendency.

- Availability. Let S be a web service. The availability of S is the probability that S can be accessed during a time cycle, denoted as $q_a(S)$. We can give the calculating formula as follow according to the definition

$$q_a(S) = \frac{N_{AS}}{N_{AT}} \quad (1)$$

where N_{AS} is the number of times that S has been successfully accessed in a statistical cycle and N_{AT} is the number of times the service has been requested in the same cycle.

Evidently availability is a QoS property with positive value tendency which means its increment will contribute to the quality of the web service. Thus on account of the thought above, we need to reverse the value tendency:

$$Q_a(S) = \frac{1}{q_a(S)} = \frac{N_{AT}}{N_{AS}} \quad (2)$$

- Execution time. Let S be a web service. The execution time of S is the time span from the moment service requester send request to the moment the execution result is received, denoted as $q_{et}(S)$ and hence be calculated by following formula

$$q_{et}(S) = T_{RT} - T_{ST} \quad (3)$$

where T_{RT} is the moment the service S returns the response and T_{ST} is the moment service requester sends the request. We use average value in a statistical cycle to present this attribute on account of the instability of network. Formula is presented as follows

$$Q_{et}(S) = \frac{\sum_{i=1}^n (T_{RT_i} - T_{ST_i})}{n} \quad (4)$$

where n is the number of times the service are accessed in the time cycle.

- Reliability. Let S be a web service. The reliability of S is the probability S returns correct execution results during a statistical cycle, denoted as $q_r(S)$. It's formulized as following expression

$$q_r(S) = \frac{N_{RS}}{N_{RT}} \quad (5)$$

where N_{RS} refers to the count the service S returns correct execution results and N_{RT} refers to the count S is requested. Obviously, this is another QoS property with positive value tendency. So we transform formula (5) to

$$Q_r(S) = \frac{1}{q_r(S)} = \frac{N_{RT}}{N_{RS}} \quad (6)$$

- Execution price. Let S be a web service. The execution price is the fee that the service requester needs to pay for using the service, denoted as $q_{ep}(S)$.

Finally on the basis of above QoS properties, the QoS of web service can be defined as

$$QoS(S) = (q_a(S), q_{et}(S), q_r(S), q_{ep}(S)) \quad (7)$$

4. Standardization of QoS Properties

In order to balance different magnitudes or ranges of QoS properties resulting from different measurement criteria, people process QoS properties by standardizing them to locate in interval [0-1]. The processing methods can be summarized to divide each property value by a fixed value. The only difference is the way of getting the fixed value. The traditional approach adopts max-min difference as fixed value and tackles the zero value problem of fixed value separately [6] which will impair the performance to some degree.

This paper uses max-min sum instead of traditional max-min difference in order to avoid judging whether max value equals to min value when standardizing the property value. And thus this will enhance the execution efficiency of algorithm.

5. Calculating QoS Value of a Web Service and Introduction of Performance-Price Ratio

After the standardization of QoS properties we still have a problem that how to use these property values to calculate QoS value of the web service. People usually use a linear weighted approach to get it:

$$QoS(S) = \sum_{i,j=1}^n W_j * q_i \quad (8)$$

where $W_j \in [0,1]$ is the weight of each QoS property and $\sum W_j = 1$; q_i represents the standardized QoS property value; n is the count of QoS properties of a web service.

The present approach to set the value of W_j in formula (8) is by subjective way. This approach will make it difficult for non-professional users to make use of this approach and adjust the weight settings when introducing new QoS properties. So we present a hybrid subjective-objective approach to set the weights based on the coefficient of variation [10, 11]. The main idea is to get relative weights by variation coefficient method and the weights will act as recommended values. And users can adjust the weight according to their own preference.

After these processing steps we can use formula (8) to calculate the QoS value of the web service. However, there is still an issue to consider that several web services may have the same max QoS value. The common method is to choose a web service randomly. We make some improvements and propose a new way by introducing performance-cost ratio. If we find that several services have the same max QoS value we continue to calculate their performance-cost ratio. This is accomplished by dividing the performance attribute (such as availability, reliability *etc.*) by price value of the service. And finally recommend the candidate web service possessing the max performance-cost ratio to service users. The complete algorithm is presented in Algorithm 1.

Algorithm 1. Description of our Algorithm

```

//standardization
For each service Si in candidate service CS do
  //standardize QoS properties of service Si
   $q'_j = [Max(q_j) - q_j] / [Max(q_j) + Min(q_j)]$ ;
End For
//set weights
For each service Si in candidate service CS do
   $V_i = \frac{\sigma_i}{q_i}$ ; //calculate coefficient of variation
   $W_{q_i} = \frac{V_i}{\sum V_j}$ ; //calculate the proportion of respective coefficient of variation of QoS
  //properties
End For
 $W'_{q_i} = \text{getinput}(w_i)$ ;
//calculate QoS value of each service
For each service Si in candidate service CS do
   $QoS(S_i) = \sum W'_j * q'_i$ ;
End For
//sort in descending order
For i=1 to m do //m: count of candidate services
  For j = i to m do
    If QoS(Si) < QoS(Sj) then
      Tmp = QoS(Si);
      QoS(Si) < QoS(Sj);
      QoS(Sj) = Tmp;
    EndIf
  End For
End For
//calculate performance-cost ratio
For i=m to 2 do //m: count of candidate services
  If QoS(Si) = QoS(S1) then
    Calculate the performance-cost values of S1...Si and sort;
  EndIf
End For
//output
Printout(Si); //output sorted candidate services

```

6. Analysis of Algorithm Complexity

The computing cost of traditional linear weighted service selecting algorithm will increase as the increasing of the quantity of candidate web services and the count of QoS property indexes. Its asymptotic time complexity equals to O(mn). The time complexity of our algorithm mainly includes four parts.

- Standardization of QoS properties value. The time complexity is O(mn).
- Calculate the QoS value of candidate services. The time complexity is O(mn).
- Sort processing of QoS values. The time complexity is O(mn).
- Sort processing of performance-cost ratio when there are several candidate services having the same maximum QoS value. The time complexity is O(mn).

In general, our algorithm and linear weighted service approximately have the same asymptotic time complexity. However, due to the improvement on standardization of QoS properties (adopt unified formula, no need to judge the property value tendency and whether the fixed value equals to zero), our algorithm has better performance than linear weighted algorithm.

7. Test and Comparison

We conducted following tests to verify feasibility of the algorithm proposed by this paper and the improved performance and effectiveness over the traditional algorithm.

- Test platform
 - Hardware environment
 - CPU: Intel(R) Core(TM) i3 2.53GHz
 - Memory: DDR3 1066MHz 2G
- Software environment
 - Operating System: Windows 7 Ultimate
 - Analytical software: Excel 2007, Matlab R2009a
- Web services set for test

In these test cases we choose QWS_Dataset Version2 as our candidate web services set. This services set is provided by Doctor Eyhab Al-Masri of University of Guelph by collecting from UDDI registry, search engine library and service web portals using crawler engine WSCE. He collects 2507 web services from this set and do test cases on them in the respects of 9 common QoS properties including response time, availability, throughput, reliability, compatibility and so on at a ten-minute interval in three days.

- Comparison algorithm

The comparison algorithm we choose is the linear weighted service selecting algorithm proposed by Zeng *et al.*, [3]. This algorithm calculates weighted sum of all the standardized property values as the evaluation of the web service. The standardization method is defined as:

$$V_{i,j} = \begin{cases} \frac{Q_j^{\max} - Q_{i,j}}{Q_j^{\max} - Q_j^{\min}}, & \text{if } Q_j^{\max} - Q_j^{\min} \neq 0; \\ 1, & \text{if } Q_j^{\max} - Q_j^{\min} = 0. \end{cases} \quad (9)$$

$$V_{i,j} = \begin{cases} \frac{Q_{i,j} - Q_j^{\min}}{Q_j^{\max} - Q_j^{\min}}, & \text{if } Q_j^{\max} - Q_j^{\min} \neq 0; \\ 1, & \text{if } Q_j^{\max} - Q_j^{\min} = 0. \end{cases} \quad (10)$$

where Q_j^{\max} is the maximum of specified QoS property and represented as $Q_j^{\max} = \text{Max}(Q_{i,j}), 1 \leq i \leq n$ while Q_j^{\min} is the minimum and represented as $Q_j^{\min} = \text{Min}(Q_{i,j}), 1 \leq i \leq n$.

Formula (9) is applied to QoS properties that have negative value tendency (which means, the lower the value, the better quality the service has). Formula (10) is applied to QoS properties that have positive value tendency.

The approach to calculate the QoS value of web services after standardization is presented in formula (8).

For the convenience of describing, we use “A₁” to represent the algorithm proposed by Zeng et al. and “A₂” to represent our algorithm.

- Verify Feasibility

In order to verify the feasibility of our algorithm, we separately choose 2, 5 and 10 web services with identical functionalities as test data set. We carry out the two algorithms on three QoS properties: response time, success rate and availability. Limited by the space, we only illustrate the experiment result of 10 services in this paper.

Table 1. Test Result of 10 Web Services

Service	Response Time	Availability	Success Rate	QoS(A ₁)	QoS(A ₂)
S1	47.27	61	62	1	0.653492620
S2	63.83	19	20	0.493303	0.392433415
S3	68.91	19	20	0.484277	0.385677018
S4	64.96	18	18	0.475386	0.363768348
S5	71.54	18	18	0.463695	0.355016951
S6	68.88	17	18	0.461603	0.345520562
S7	70.23	17	18	0.459204	0.343725063
S8	143.65	19	19	0.346931	0.278930753
S9	221	18	18	0.198129	0.156235231
S10	328.67	21	21	0.040909	0.066831502

From above test result in Table 1, we can get the order of QoS values of these ten services: S1>S2>S3>S4>S5>S6>S7>S8>S9>S10 and it indicates that our algorithm based on QoS is feasible.

- Performance Comparison

As we describe above, the improvements of our algorithm over traditional one are mainly in two aspects:

- We use max-min sum as the fixed value instead of their difference value to avoid judging whether divisor equals to zero. Therefore the process of standardization is simplified.
- We get rid of the distinction of value tendency in the stage of definition to make sure that all the QoS property values have the same calculating method when standardizing them.

In order to verify the improvement of our algorithm on performance, we conduct the performance comparison test. We separately choose 10, 50 and 100 web services to conduct this comparison test on 3, 4 and 5 QoS properties(response time, availability and reliability for three-property test; plus success rate for four-property test; plus delay time for five-property test) and record the execution time of these two algorithms. Limited by the space, we only show the figure that illustrates the test of 100 services with 5 properties.

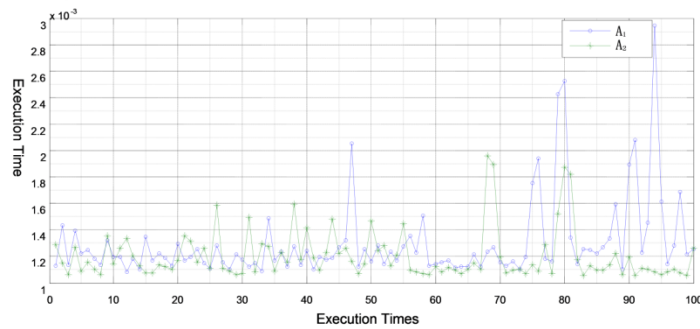


Figure 1. Execution Time Comparison of Two Algorithms

As we can see in Figure 1, our algorithm A_2 has a better performance than A_1 . It indicates that our improvements on the traditional algorithm are effective.

Table 2 shows the final test results of different number of services on different count of properties. Limited by the space, we use the average value of the statistical data to present the experiment result. This can also reduce the impact on execution time introduced by the change of system available resources.

Table 2. Performance Comparison of Two Algorithms under Various Conditions (time unit: ms)

Number of Services	Execution Time (3 properties)		Execution Time (4 properties)		Execution Time (5 properties)	
	A_1	A_2	A_1	A_2	A_1	A_2
10	689	645	871	829	1106	1071
50	701	661	894	848	1183	1123
100	745	689	938	895	1301	1204

* A_1 : Zeng's Algorithm A_2 : our Algorithm

From above results in Table 2, we can discover that our algorithm reinforces overall evaluation over Zeng's web service selecting algorithm by about 10% on the average, which indicates the improvements we adopt on the traditional algorithm are practically feasible.

8. Conclusion

This paper proposes a QoS-based calculating method of selecting web services. At present it is common to use linear weighted approach to calculate QoS value of web services. We mainly propose two improvements on the standardization process of this approach: get rid of the distinction of value tendency when defining QoS properties; use max-min sum of QoS property value when calculating the fixed value to avoid dealing with the problem that whether the divisor equals to zero. The comparison experiments indicate that our algorithm is feasible and effective and have better performance compared with traditional linear weighted algorithm. Furthermore, we also introduce hybrid subjective-objective approach based on the

coefficient of variation upon the setting of the weights of QoS properties. This improvement will make the service selecting functionality user-friendly and convenient.

With the development web service technology, composite web services have attracted people's attention. This paper's content aims at single web service so our further research work will involve applying the algorithm proposed in this paper to the selection of composite web services.

Acknowledgments

The authors would like to thank to the anonymous reviewers and their valuable suggestions to improve the quality of this paper.

References

- [1] J. O. Sullivan, D. Edmond and A. ter Hofstede, "What's in a Service", Distributed and Parallel Databases, vol. 12, no. 2-3, (2002), pp. 117-133.
- [2] A. Sheth, J. Cardoso, J. Miller and K. Kochut, "QoS for Service-oriented Middleware", Proceedings of the Conference on Systemics, Cybernetics and Informatics, Orlando, FL, (2002) July.
- [3] L. Z. Zeng, B. Benatallah, A. H. H. Ngu, M. Dumas, J. Kalagnanam and H. Chang, "QoS-aware Middleware for Web Services Composition", IEEE Transactions on Software Engineering, vol. 30, no. 5, (2004), pp. 311-327.
- [4] C. F. Lin, R. K. Sheu, Y. S. Chang and S. M. Yuan, "A Relaxable Service Selection Algorithm for QoS-based Web Service Composition", Information and Software Technology, vol. 53, no. 12, (2011), pp. 1370-1381.
- [5] K. C. Lee, J. H. Jeon, W. S. Lee, S. H. Jeong and S. W. Park, "QoS for Web Services: Requirements and Possible Approaches", <http://www.w3c.or.kr/kr-office/TR/2003/ws-qos/>.
- [6] G. S. Kang, J. X. Liu, M. D. Tang, X. Q. Liu and K. K. Fletcher, "Web Service Selection for Resolving Conflicting Service Requests", 2011 IEEE International Conference on Web Services(ICWS), Washington, DC, (2011) July.
- [7] D. A. Menascé, "QoS Issues in Web Services", IEEE Internet Computing, vol. 6, no. 6, (2002), pp. 72-75.
- [8] D. A. Menascé, "Composing Web Services: A QoS view, IEEE Internet Computing", vol. 8, no. 6, (2004), pp. 88-90.
- [9] L. Chung, "Representation and Utilization of Non-Functional Requirements for Information System Design", Advanced Information Systems Engineering, vol. 498, (1991), pp. 5-30.
- [10] A.G. Bedeian and K. W. Mossholder, "The Use of the Coefficient of Variation as a Measure of Diversity", Organizational Research Methods, vol. 3, no. 3, (2000), pp. 285-297.
- [11] M. Gulhar, B. M. Golam Kibria, A. Albatineh and N. U. Ahmed, "A Comparison of Some Confidence Intervals for Estimating the Population Coefficient of Variation: A Simulation Study", Statistics and Operations Research Transactions, vol. 36, no. 1, (2012), pp. 45-68.
- [12] S. S. Yau and Y. Yin, "QoS-Based Service Ranking and Selection for Service-Based Systems", IEEE International Conference on Services Computing, Washington, DC, (2011) July.
- [13] H. Jin, H. Zou, F. C. Yang, R. H. Lin and X. C. Zhao, "QoS-based Service Selection for Multiple Requests with Different Classes", 2012 9th International Conference on Service Systems and Service Management(ICSSSM), Shanghai, (2012) July, pp. 604-609.

Authors



Yuqiang Li is an associate professor of the Department of Computer Science at Wuhan University of Technology, China. He received his Ph.D. degree in Computer Application Technology from Wuhan University of Technology. His research interests include big data processing and cloud computing.



Wei Zhao is currently a postgraduate student of the Department of Computer Science, Wuhan University of Technology, China. He received his Bachelor's degree from Wuhan University of Technology. His research focuses on high-performance data access through parallel computing.



Lei Che is a lecturer at the School of Information and Management, Beijing Information Science and Technology University. She received her Master Degree in Computer Application Technology in 2005 from Wuhan University of Technology, and she is studying for doctor's degree in Computer Application Technology at Renmin University of China. She is the leader of The Youth Talent Project of Beijing Universities(YETP1503). Her research focuses on Data Base, Data Mining and Business Intelligence.