Journal of Applied Sciences

ISSN 1812-5654

Journal of Applied Sciences 13 (24): 5734-5738, 2013
ISSN 1812-5654 / DOI: 10.3923/jas.2013.5734.5738

A Service Selection Method Based on Web Service Clusters

[1,2]Wei Liu, [1]Yu-Yue Du and [1]Chun Yan
[1]College of Information Science and Engineering, Shandong University of Science and Technology, 266590, Qingdao, China
[2]The Key Laboratory of Embedded System and Service Computing, Ministry of Education, Tongji University, 200092, Shanghai, China

Abstract: Service selection is an essential issue in service discovery based on Web service clusters. The ratio of trust value to price is introduced to select the most suitable service in a cluster. The method of computing the initial trust, the direct trust and the recommendation trust are, respectively presented and the overall trust is derived based on them. The proposed method keeps the balance between the price and service quality.

Key words: Web service clusters, service selection, trust, QoS

INTRODUCTION

Trust model and computation have been discussed in many studies The knot model (Gal-Oz *et al*., 2008) calculates trust on the ground of a subset of the community members. Eigen-trust (Kamvar *et al*., 2003) is based on transitive trust. Bayesian estimation methods are adopted to get trust based on transactions and shared experiences in P2P systems (Wu and Wu, 2009). A trust model of Web services composition is presented on the basis of a Bayesian formalization of the trust. The trust is changed according to experiences with Web services (Paradesi *et al*., 2009). The reputation of a party is attained on the basis of beta distribution with positive evidence and negative evidence as two parameters in the Beta reputation model (Josang and Ismail, 2002). A method of selecting resource sites meeting a customer requirement is formulated based on the trust. The problem is solved by optimizing within the limitations of Service Level Agreement (SLA) (Xiong and Perros, 2006). RATE-Web is given for a reputation model of Web services to be used for selecting Web services on the ground of trust. Experiences of service providers can share with their partners by means of feedback ratings (Malik and Bouguettaya, 2009). A reputation model of Web services is put forward to consider the arrival of requests and discuss the impact on the overall reputation (Khosravifar *et al*., 2010). Trust and reputation have also become key components of several commercial systems such as E-bay.

In this study, for paid Web services based on Web service clusters, a service selection method is given. Service clusters are divided based on the function of each service. A Web cluster is a set of services with the same function. If a service requestor presents a service request, the service cluster first is matched according to the function. If a service cluster is chosen, which service in the service cluster is selected to execute the service request? For paid Web services, each Web service has its price. To keep the balance between the price and service quality, a web service is selected based on the ratio of trust value to price. Based on the request which is taken as cooperation between the service and the service requestor, the service with the highest ratio is selected to execute. The ration is proportional to trust value and inversely proportional to the price. Hence, to gain the execution chance, a service need increase its trust value and decrease its price.

Overall trust is composed of three parts: The initial trust, direct trust and recommendation trust. When a service registers in a registration center, it is assigned to a service cluster with the similar function. It has no trust value and thus cannot become a candidate service to participate in competition for a service request. Therefore, an initial trust need to be given to start a new service to participate in competition. We calculate by the test value of QoS (Quality of Service) during the testing period. When a service in a service cluster is selected, it is executed to complete the service request. After the service is executed, QoS is computed according to the execution. The direct trust value is derived and updated based on QoS after the service is executed. If the difference between the current QoS and the published QoS at the registration center is small, the direct trust is increased to show reward. If the difference between the current QoS and the published QoS at the registration center is beyond a certain range, the direct trust is decreased to show punishment. Since each Web service has its price, to gain the execution chance, a service needs to ensure

Corresponding Author: Wei Liu, College of Information Science and Engineering, Shandong University of Science and Technology, 266590, Qingdao, China

the quality of service. Besides, the recommendation trust needs to be considered to calculate the overall trust. We use norm grey correlation analysis method to calculate recommendation trust of Web services in Liu *et al*. (2013).

COMPUTATION OF QoS

QoS is an aggregate indicator used to evaluate the degree of satisfaction after using a Web service. The main quality attributes to service requestors are response time, throughput, availability and accessibility, etc. Some of them are positive attributes while some of them are negative attributes. The larger a positive attribute is, the better the quality of services is. In other hand, the smaller a positive attribute is, the worse the quality of services is. So, the equation of response time and availability are given as follows.

Response time: Response time denotes the interval from the time of sending the service request to the time of receiving the service response:

$$\text{Time}(s) = \frac{\sum_{i=1}^{n} \text{Receive}_i(s)\text{-Send}_i(s)}{n} \tag{1}$$

where, Receive_i (s) represents the response time of the ith calling for a Web service s and n denotes the number of times of calling for s.

Availability: Availability denotes the success probability of calling for Web services:

$$\text{Avaliability}(s) = \frac{M(s)}{N(s)} \tag{2}$$

where, M (s) represents success times of calling for a Web service s and N (s) denotes the total number of times of calling for s.

There are still many quality attributes. The equation of other quality attributes can refer from related references.

When choosing a service according to service quality, multiple service attributes can be regarded as a whole to compute an overall service quality. Because there are differences among the presentation and quantization of different service quality attributes, different quality attributes of Web service need to be converted into proper dimensionless indexes. Positive attributes are transformed into dimensionless indexes with the following equation:

$$R=\begin{cases} \dfrac{Q\text{-}Q_{min}}{Q_{max}-Q_{min}} & \text{if } Q_{max}-Q_{min}\neq 0 \\ 1 & \text{if } Q_{max}-Q_{min}=0 \end{cases} \tag{3}$$

Negative attributes are transformed into dimensionless indexes with the following equation (Wu and Wu, 2009):

$$R=\begin{cases} \dfrac{Q_{max}\text{-}Q}{Q_{max}-Q_{min}} & \text{if } Q_{max}-Q_{min}\neq 0 \\ 1 & \text{if } Q_{max}-Q_{min}=0 \end{cases} \tag{4}$$

where, Q_{min} represents the maximum values of a service quality attribute while Q_{min} is the minimum value. Q denotes the concrete value of a service quality attribute. The equation of the overall service quality is as follow:

$$\text{IntegrativeQos} = \sum_{i=1}^{r} w_i q_i \tag{5}$$

where, r represents the number of service quality attributes, W_i is the weight of a service quality attribute and q_i denotes the dimensionless index of a service quality attribute. The overall service quality is a comprehensive criterion for evaluating service quality.

COMPUTATION OF TRUST AND WEB SERVICE SELECTION

Initial trust value: When a new Web service is published and registered to a registration center, an initial trust value is added, or else a new Web service cannot compete for being selected. To solve the problem, the parameter initialization method is used in some literatures. However, there exist many drawbacks. In this study, the initial trust value is calculated based on the performance of Web services during testing period. The initial trust value of Web services depends on the honesty of a Web service provider. The honesty is obtained by the difference between the real quality attribute value during the testing period and the quality attribute value published by a Web service provider.

The quality attribute value published by a Web service provider can be determined base on the Service Level Agreement (SLA). When a Web service provider publishes a new Web service to a registration center, the data concerning the quality of Web services should be published by the service provider. When a new Web service s_i is registered to a registration center, the initial trust value of s_i is computed as follow:

$$T_i^0 = \frac{\sum_{k=1}^{r} q_i^{kp} / q_i^{ka}}{r} \quad \text{if } q_i^{kp} > q_i^{ka},\ q_i^{kp} = q_i^{ka} \tag{6}$$

where, r represents the number of service quality attributes, T_i^0 is the initial trust value of a new service, q_i^{ka} is the kth quality attribute value of Web service s_i published by the service provider. q_i^{kp} denotes the test values of the kth quality attribute of Web service s_i during the testing period.

Update of direct trust values: When service requestors send a request, a Web service is used by a service requestor and a history record is produced. A history record includes information about some operation situation and QoS of a Web service. In this case, a service requestor gets the trust value for a concrete Web service according to history records produced after the service is used by the service requestor.

Suppose $Q(q_1, q_2,...q_r)$ is an attribute set of QoS obtained after a concrete service s_i implements the service request where $0 \le q_j \le 1$, $1 \le j \le r$, r represents the number of service quality attributes, q_j denotes dimensionless values of QoS of a Web service. After a service s_i completes a service request, reward or punishment is given based on results. Whether it is executed successfully or is determined by the difference between the actual quality attribute values after a service s_i completes a service request and the registered quality attribute values published by the service provider in registration center. Let $\varepsilon \in [0, 1]$ be a threshold for deciding whether the execution is successful or not:

$$\left(\sqrt{\frac{\sum_{k=1}^{r} w_k (q_j^{kx} - q_j^{ka})^2}{r}} \le \varepsilon \right)$$

denotes the successful execution and otherwise unsuccessful execution. If it is a successful execution, the service s_j is rewarded and the trust value is increased. A_j^x is defined to denote the increased trust value obtained by service s_j after the xth successful execution where $0 \le A_j^x \le 1$, $1 \le x \le ns$ and ns denotes successful execution times. A_j^x is computed as follow:

$$A_j^x = 1 - \sqrt{\frac{\sum_{k=1}^{r} w_k (q_j^{kx} - q_j^{ka})^2}{r}} \quad \text{if } q_j^{kx} > q_j^{ka},\ q_j^{kx} = q_j^{ka} \tag{7}$$

where, q^{kx}_j is the kth quality attribute value of a service s_j on the xth successful execution and w_k is the weight the of kth quality attribute.

If it is an unsuccessful execution, the service s_i is punished and the trust value is decreased. P^y_i is defined to denote the decreased trust value obtained by service s_i after the yth unsuccessful execution where $1 \le y \le nf$ and nf denotes unsuccessful execution times. P^y_i is computed as follow:

$$-\sqrt{\frac{\sum_{k=1}^{r} w_k (q_i^{ky} - q_i^{ka})^2}{r}} \tag{8}$$

where, q^{ky}_i the kth quality is attribute value of a service s_i on the yth unsuccessful execution and w_k is the weight the kth quality attribute. The larger the difference is, the larger the punishment and the decrease of the trust value. Otherwise, the smaller the difference is, the smaller the decrease of the trust value is.

In addition, the trust is dynamic attenuation as time passed. The earlier trust fades as time passed. The new trust during the recent interactions represents better the present trust relationship. We take the attenuation function as the weight of each cooperation to embody such change. The attenuation function is defined as follow:

$$\lambda(j,t_j) = \begin{cases} 1 & j=l, t_j = t_l \\ \lambda(j+1, t_{j+1}) \frac{t_{j+1} - t_j}{t_l - t_1} & 1 \le j \le l-1, t_1 \le t_j \le t_{l-1} \end{cases} \tag{9}$$

where, $j \in [1, l]$ represents the jth cooperation, l is the total number of cooperation in history. $t_j \in [t_1, t_l]$ denotes the time of the jth cooperation $\lambda(j, t_j) \in [0, 1]$. The last time the cooperation happen, the attenuation function is assigned as 1 denoting no attenuation. The earlier the cooperation is, the smaller the value of the attenuation function which shows that the weight for computing the trust value decreases.

The method to calculate the trust value concerning time interval and cooperation times reflects the dynamic properties that trust varies with time. Whether the cooperation is successful or not every time affects the computation of the weight of not only this time but also each time previously. When the cooperation is successful at last time, the total trust value increases and the weight of previous unsuccessful cooperation relatively decreases which embodies the principle that encourages success. When the cooperation is unsuccessful at last time, the total trust value decreases and the weight of previous successful cooperation relatively decreases which embodies the principle that publishes failure.

In conclusion, the equation of direct trust value is adopted as follow concerning the cooperation history, successful and unsuccessful cooperation and dynamic change of time:

$$T_i^1=\begin{cases}0 & ns,nf=0\\ \dfrac{\sum_{x=1}^{ns}\left[A_i^x\times\lambda(x,t_x)\right]}{ns} & ns>0,nf=0\\ \dfrac{\sum_{y=1}^{nf}\left[P_i^y\times\lambda(y,t_y)\right]}{nf} & ns=0,nf>0\\ \dfrac{\sum_{x=1}^{ns}\left[A_i^x\times\lambda(x,t_x)\right]+\sum_{y=1}^{nf}\left[P_i^y\times\lambda(y,t_y)\right]}{ns+nf} & ns,nf>0\end{cases}\quad(10)$$

When the times of cooperation in history are much more, the computation time becomes longer. To reduce computation time, we use a threshold of a moving window num which defines the recent cooperation times used to compute the direct trust value. If the times of cooperation in history $j\leq num$, the above equation is used to compute the direct trust value. Otherwise, if $j>num$, the direct trust value is calculated based on the recent num cooperation times and the equation is as follow:

$$T_i^1=\begin{cases}0 & sn,fn=0\\ \dfrac{\sum_{xn=1}^{sn}\left[A_i^{xn}\times\lambda(xn,t_{xn})\right]+T_i^{1-num+1}\lambda(j-num,t_{j-num})}{num} & sn>0,fn=0\\ \dfrac{\sum_{yn=1}^{fn}\left[P_i^{yn}\times\lambda(yn,t_{yn})\right]+T_i^{1-num+1}\lambda(j-num,t_{j-num})}{num} & sn=0,fn>0\\ \dfrac{\sum_{xn=1}^{sn}\left[A_i^{xn}\times\lambda(xn,t_{xn})\right]+\sum_{yn=1}^{fn}\left[P_i^{yn}\times\lambda(yn,t_{yn})\right]+T_i^{1-num+1}\lambda(j-num+1,t_{j-num+1})}{num} & sn,fn>0\end{cases}\quad(11)$$

where, sn denotes the recent num-1 successful cooperation times, A^{xn}_i is the recent num-1 unsuccessful cooperation times, P_i^{yn} represents the xnth successful cooperation of the recent num-1 successful cooperation times, $\lambda(xn, t_{xn})$ is the ynth unsuccessful cooperation of the recent num-1 unsuccessful cooperation times, $\lambda(yn, t_{yn})$ is the attenuation factor of the ynth unsuccessful cooperation of the recent num-1 unsuccessful cooperation times.

Recommendation trust: We use the method to calculate recommendation trust of Web services like Liu *et al*. (2013). The weight of every recommender is confirmed by norm grey correlation analysis method. The method quantifies the weight of service recommenders, avoids the vicious recommendation and computes the recommendation trust value of each service improving the reliability of selected services. We described the method in detail in Liu *et al*. (2013) and don't dwell on it in this study.

Overall trust: The overall trust is calculated as follow. Assume a service requestor makes a request req and $S = \{s_1,s_2,\ldots,s_n\}$ is a service set satisfying the request. $T_{sj}^{\,0}$ represents the initial trust of service s_j, $1\leq j\leq n$. $T_{sj}^{\,d}$ is the direct trust of service s_j. $T_{sj}^{\,r}$ denotes the recommendation trust of service s_j. The equation of the overall trust $T_{sj}^{\,a}$ is defined as follow:

$$T_{sj}^a=\omega_1T_{sj}^0+\omega_2T_{sj}^d+\omega_3T_{sj}^r\quad(12)$$

where, $\omega_1+\omega_2+\omega_3 = 1$, ω_i, i = 1, 2, 3 denotes the weight of the initial trust, the direct trust and the recommendation trust.

Web service selection: Let $P = \{p_1, p_2, \ldots, p_n\}$ be a set of each service price. p_j, $1\leq j\leq n$, denotes the price of service s_j. The ratio of trust value to price is defined as follow:

$$R=\frac{p_j}{T_{sj}^a}\quad(13)$$

The service with the highest ratio is selected to execute the request. The ratio is proportional to trust value and inversely proportional to the price. Hence, to gain the execution chance, a service needs increase its trust value and decrease its price.

CONCLUSION

In a service cluster, each service belonging to a same service cluster has the similar function. To choose an appropriate service to implement the service request, the ratio of trust value to price is used. The initial trust of a service is obtained by the difference between the real quality attribute value during the testing speriod and the quality attribute value published by a Web service provider. The update of direct trust values is based on history records and produced after the service is executed by the service requestor. In computing the direct trust, to describe the trust is dynamic attenuation as time passed, the attenuation function is taken as the weight of each cooperation. To reduce computation time, a threshold of a moving window is introduced to express the recent cooperation times used to compute the direct trust value. The recommendation trust is calculated based on norm gray correlation analysis method. The overall trust is achieved according to the above three kinds of trust. The service with the highest ratio of trust value to price is selected to execute the request.

ACKNOWLEDGMENTS

This study is supported by the National Basic Research Program of China with Grant No. 2010CB328101; the National Natural Science Foundation of China with grant number 61170078; The Doctoral Program of Higher

Education of the Specialized Research Fund of China with Grant No. 20113718110004; The Scientific and Technological Developing Program of Shandong Province of China with Grant No. 2011 GGX10114; The Project of Shandong Province Higher Educational Science and Technology Program with Grant No. J12LN11 and the SDUST Research Fund of China with Grant No. 2011 KYTD102, 2011KYTU104.

REFERENCES

Gal-Oz, N., E. Gudes and D. Hendler, 2008. A robust and knot-aware trust-based reputation model. Proceedings of IFIPTM 2008: Joint iTrust and PST Conferences on Privacy, Trust Management and Security, June 18-20, 2008, Trondheim, Norway, pp: 167-182.

Josang, A. and R. Ismail, 2002. The beta reputation system. Proceedings of the 15th Bled Electronic Commerce Conference on e-Reality: Constructing the e-Economy, June 17-19, 2002, Bled, Slovenia, pp: 41-55.

Kamvar, S.D., M. T. Schlosser and H. Garcia-Molina, 2003. The eigentrust algorithm for reputation management in p2p networks. Proceedings of the 12th International Conference on World Wide Web, May 20-24, 2003, Budapest, Hungary, pp: 640-651.

Khosravifar, B., J. Bentahar and A. Moazin, 2010. Analyzing the relationships between some parameters of web services reputation. Proceedings of the 8th IEEE International Conference on Web Services, July 5-10, 2010, Miami, FL., USA., pp: 329-336.

Liu, W., Y.Y. Du and C.H. Yan, 2013. A method to calculate recommendation trust of Web services. J. Software Eng.,

Malik, Z. and A. Bouguettaya, 2009. RATEWeb: Reputation assessment for trust establishment among web services. VLDB J., 18: 885-911.

Paradesi, S., P. Doshi and S. Swaika, 2009. Integrating behavioral trust in web service compositions. Proceedings of the 7th International Conference on Web Services, July 6-10, 2009, Los Angeles, CA., USA., pp: 453-460.

Wu, P. and G. Wu, 2009. A reputation-based trust model for P2P systems. Proceedings of the International Conference on Computational Intelligence and Security, December 11-14, 2009, Beijing, China, pp: 352-356.

Xiong, K. and H. Perros, 2006. Trust-based resource allocation in web services. Proceedings of the IEEE International Conference on Web Services, September 18-22, 2006, Chicago, USA., pp: 663-672.