A Web Service Organization Method based on Cloud Cluster and Cloud Service Community

Pengfei Zhao,Ruzhi Xu,Zhikun Zhao,Peiyao Nie
School of Computer Science and Technology
Shandong University of Finance and Economic
Jinan, China
goobycle@126.com

Bo Fan
Department of Information Technology
Shandong Energy Group Co.Ltd
Jinan, China
ruzhixu@sdfi.edu.cn

***Abstract*—against the lower searching efficiency and lower precision existing in the web service discovery system, this paper puts forward a web service organization framework based on cloud service community. The community exists as a form of net built by the web services (nodes) and the cooperative relationships (edges) among them. And the web services can be replaced by others which has the same function but different quality. Thus the framework also clusters all of the services which have the similar function by similarity matching algorithm. Through the service selecting algorithm, the community can offer the best services that user required even the potential demanding ones. In addition, Experimental results show that the improved service organization framework can promote and precision ratio while the service discovery efficiency also be raised greatly.**

Keywords-similarity matching algorithm; service organization form; cloud service community

I. INTRODUCTION

In the environment of cloud computing, users can access web service easily by SOA (Service-Oriented Architecture), and through which, the high interoperability, cross-platform and loose coupling characteristics of Web Service fully appears. But there are some problems, such as low matchmaking degree and poor mechanism of dynamic organization are still limiting the efficiency and precision. But, researchers have done a lot of works for improving the results on service matchmaking, service clustering and other aspects.

The first aspect is service matching. The precision of similarity matching methods is higher from that based on key words to semantic ontology. These methods can be divided into four classes, namely method based on distance, content of information, attributes and mixing. From these methods, researchers put forward many web service similarity matching methods, such as function matchmaking and process matchmaking. But less took the relevance among services and relationship among attributes into consideration.

Utilizing the features of service matchmaking, lots of attempts has been taken to strengthen the web service discovery framework. These researchers increased the service discovering efficiency by improving service clustering [2] style or organization model, even against different roles of SOA. For example, user driven service clustering method [3], which sends the matching result to local zone according to user's service description, organizes the returned services into a service pool, packs services into interface in order to provide no difference service for users, can be seen as the prototype of service clustering. But it aggravates the burdens of client and impacts no good on service quality. The P2P discovering model of semantics and double layer constructor in [4, 5] proposed the concept of ontology community and organized the web services in the community by the organizational form of P2P. It has a good extension and satisfies the requirement of big data. But it can't match the efficiency of centralized one. The clustering method based on spanning tree in [11] distributes the duty of registering center to clustering tree for releasing the burden of center. But it increased the burden of super node and has no good in system stability and lowered the precision by matching on the standard of super node.

Considering all of the above, this paper mainly investigated the approach of improving service organization framework, and presents a new one, which organizes the services as cloud service community. Clustering process happens on the side of cloud in order to get stronger and more extensive. Cloud service community connected the cooperative services to recommend relevance service. As a result, the framework can increase service discovering efficiency and precision.

II. CLOUD CLUSTERING FRAMEWORK

Service similarity matchmaking is vital in service discovering and is the key whether the clustering is good or not. There is also relevance existing among web services on function description besides similarity. The similarity degree of services is the weighted sum of each parameter with different importance while Boolean predicate is added according to the inner relationship among the parameters. The method putting the registering center on the cloud can resolve the bottleneck problem that centralized one has and improve discovering efficiency.

A. Semantic similarity matchmaking algorithm

The WSOSM (Web Service Ontology Semantics Machine) studied in this paper has two functions. One the function is similarity matchmaking based on ontology and

Published by Atlantis Press, Paris, France.

another is service relevance matchmaking based on ontology related to domain. The similarity laids emphasis on functional mutual replacement among services, but relevance puts more attentions on cooperative relationship.

Definition 1 The semantic similarity between concept A and B is defined by similarity matchmaking method based on domain ontology (tree structure). The calculation formula is as follows.

$$sim(A,B)=\begin{cases}\dfrac{\alpha\times(D(A)+D(B))}{(dist(A,B)+\alpha)\times D_{\max}\times\max(|D(A)-D(B)|,1)} & A\neq B\\ 1 & A=B\end{cases}\qquad（1）$$

$sim(A,B)$ means the semantics similarity between A and B. $D(A)$ means the height of A in the ontology tree. $dist(A,B)$ shows the shortest path of two concepts in the tree, which is also called ontology distance. $D_{\max}$ is the deep of ontology tree. $\alpha\geq 0$ is greater than or equal to 0 and is a tunable parameter.

Semantic similarity matchmaking is mainly applied to web service clustering comparing from aspect of function description, operation, input parameter and output parameter. The function description similarity matchmaking is calculated by (1) formula directly. In consideration of the one-to-many relationship between web service and operation or IO parameters, we make matches among parameters with the same type by the strategy of bipartite graph. Presume $OP_A=\{op_{a1},op_{a2}\cdots,op_{am}\},\ \ m>0$, $OP_B=\{op_{b1},op_{b2}\cdots op_{bm}\},n>0$ and $sim(op_{ai},op_{aj})\geq\beta(\beta>0)$. If and only if $sim(op_{ai},op_{aj})\geq\beta(\beta>0)$, the operation similarity formula of $op_{ai}\approx op_{aj}$ is as follows.

$$sim(<OP_A,OP_B>)=\frac{2\times\sum_{i=0,\,j=0}^{m\times n}(sim(op_{ai},op_{aj})\geq\beta?1:0)}{m+n}\qquad（2）$$

With a view to that if $similarity(<O_A,O_B>)=0$, even $similarity(<I_A,I_B>)=\infty$ is also insignificance. The relevance between output parameters and input ones should be taken into consideration in the comparison of two parameters. Using formula (2), the formula of input and output parameters similarity matchmaking with relevance is as follows:

$$similarity(<IO_A,IO_B>)=\mu\times sim_{OP}(<O_A,O_B>)(\mu\times sim_{OP}(<O_A,O_B>))+\upsilon\times sim_{OP}(<I_A,I_B>)\qquad(3)$$

$\mu,\upsilon>0$ and is assigned as the importance of input and output parameters.

In conclusion, we have gotten the similarity calculating formulas of function description, operation and IO parameters. Weight and sum them, the similarity value of two web services can be worked out. We resume that $\iota,\kappa,\lambda>0$ and they are the weight of sim_D 、 sim_{OP} and sim_{IO} and $\iota+\kappa+\lambda=1$. The formula is as follows.

$$sim_{ws}(<ws_A,ws_B>)=\iota\times sim_D+\kappa\times sim_{OP}+\lambda\times sim_{IO}\qquad（4）$$

B. Clustering framework

We are using the spanning tree as the form of services clustering and adding the feature of cloud computing into the process with the attention of facilitating service discovering. The framework can be divided into SRC (Service Register Center), CRC (Cluster Register Center) and ws (Web Service Node). SRC is the entrance of both service registering and service applying which has the record of all the information of CRC. It works on matching the registering services to CRC using WSOSM. SRC hands the part registering, organizing and service selecting over to CRC. CRC is the center of similarity services. It mainly checks quality pedigree, guarantees validity of nodes, deletes invalid nodes and most important of all it selects best services by service selecting algorithm according to the QoS parameters that user submits. CRC is produced dynamically by SRC as needed. The first service that applies to construct CRC is called Benchmark Service (BS). BS' functional description would be used when WSOSM matches services with CRC.

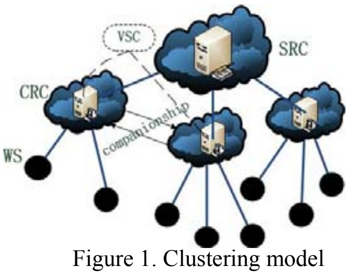

Figure 1. Clustering model

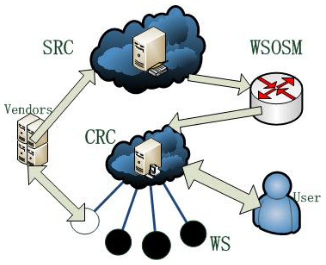

Figure 2. service registering and

WSOSM can be seen as a semantic machine that is inputting web service and outputting CRC while it exists as a form of algorithm. WSOSM accepts two types of inputting parameters referring to two different functions. One is service clustering and another is services with CRC by the user's query apply. The process of registering and applying is showed in the figure 2. Take registering for example, service provider first sends application of service registering to SRC. WSOSM calculates the CRC (or builds new one) with service clustering algorithm when it gets the service parameters given by SRC by which the service is checked. The applier will get reply after the further checking by CRC.WSOSM presets minimum threshold $\delta_{\min}$ of similarity and matches the service that SRC provided with all the

Benchmark Services in the CRC successively in order to get the biggest matching degree $\delta_{\max}$ and best matching CRC---$CRC_{\max}$. If the biggest degree matches the lowest requirement, the service will be added into the $CRC_{\max}$. Else a new CRC will be constructed taking the service as the Benchmark Service. The algorithm of service applying is much similarity to clustering. The whole process of clustering is executing on the cloud for using the centralized feature of cloud computing to improve service government efficiency and avoid the network cost of distributed one and using the dynamic extensive feature of cloud to reduce unnecessary resource-wasting.

CRC also works on monitoring whether service node exists or not. Once the service provider can't be connected, be removed will the related service node. CRC exists because of web services. All the service besides Benchmark Service meet the condition of $sim_{ws}(< ws, CRC_i >) > \delta_{\min}$. And the clustering process can make sure that the relationship between inner service and Benchmark Service in this CRC is closer than others. When the number of services in the CRC reduces to 0, CRC will cancel itself.

III. Service organizing method based on VSC

A. Definition

Definition 2 Web companionship is the cooperation relationship between services as they are focusing on same domain or have much higher composing frequency. The companionship can be presented as quintuple form.

$$companionship < ws_i, ws_j, Direction, Intimacy, profile >$$

Direction shows whether the relationship of the two web services has direction or not. Intimacy is intimacy degree between the two services. Profile is the description of relationship.

Priority of relevance between services is lower than similarity which can be translated to if $ws_i \in CRC_x$ and $ws_j \in CRC_x$, then $\exists(ws_i \in VSC_y \wedge ws_j \in VSC_y)$.

VSC is a kind of graphic structure composed by different services and the cooperation relationships between services and it exists with the relationship. The concept of VSC can be is defined as follows.

$$VSC = (WS, Companionship)$$

In which, WS is a finite and not null set and Companionship is the finite set of pairs between web service nodes. The higher of degree of relevance between services, the more possibly cooperating and more definitude the tendency of whole VSC will be.

VSC and its inner web services obey four rules as follows.

Rule 1 A web service is only owned by one VSC.

Rule 2 Resume that the intimacy between services in VSC is ω if and only if $Intimacy(ws_i, ws_j) \geq \omega$, then $companionship < ws_i, ws_j, Direction, Intimacy, profile > \in VSC_x$ 。

Rule 3 The form of VSC is organized as graphic, but the relationship between nodes is two-dimensional so that the whole VSC can be traversed from each node.

Rule 4 The cooperating relationships in the VSC divided into directed and undirected and in which directed one suggests that there is a flow among services and undirected one shows the same focus must exist among services, such as address.

The constructor of VSC has decided the style of management to be distributed which means that the adding and deleting of member service can be completed dynamically. When cooperation relationship was produced between ws_i and ws_j and it meets rule 2, the relationship--$companionship < ws_i, ws_j >$---will be saved in each of the two nodes. Unless the one of the related service is removed, the relationship will be live all the time.

B. VSC generating algorithm

The generating of VSC that consists of cooperative relationship is the constructing process of relationship. There are two ways as follows.

(1) Generated from service registering. The pattern mainly depends on judging domain attributes of service by WSOSM.

(2) Generated based on the historical data of services composing. The required services are gotten through semantics analysis when user submits requests. The degree of intimacy will be strengthened according to the fact that each service in the stream would complete a whole logic process. When the accumulative value comes to the threshold that was preset, the cooperative relationship will be constructed, in which directed relation is usually produced now.

The forming process of the first generating pattern is illustrated as follows. The generating process is taking place during service registering when WSOSM matches the applying service and Bench Service and decides whether putting the registering service into the VSC where the Bench Service exists or not. The whole process is shown in Figure 3. Resume that registering service is ws, the Bench Service of CRC and the threshold of constructing cooperation relationship is θ.

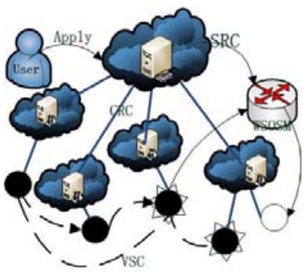

Figure 3. Process of service's joining the VSC

The second pattern is based on the service composing. When two services occur together in lots of compositions, it means that the two must have some cooperative relationship. For adjusting the increasing intimacy, $map < ws_i, intimacy >$ was set to count intimacy. Resume that time interval is Δ, applying user is A and the applying service is ws_i. The algorithm is summarized as follows.

Step 1 From now t_i forward Δ, get t_j. The relationship between services applied from t_j to t_i and ws_i is closely and they have some cooperating relationship. If ws_i and ws_k are not in the same CRC, the intimacy would be added one--- $map.put(ws_k, map.get(ws_k)+1)$.

Step 2 $If \quad Intimacy(ws_i, ws_k) \geq \omega, \quad then$ $companionship < ws_i, ws_k, Dirction, Intimacy, profile > \in VSC_x$.

C. Simple application of VSC.

Analyzing the request that user submits, each specific service can be gotten. Best services that CRC recommends are easily gotten by the service applying algorithm. Although the multi-objective programming problem of service composing is NP hard, the CRC is much benefit for seeking the optimum solution by theory of genetic algorithms. The used schemes that are comparatively maturely have a high indicative value to most users in spite that the service requirements of different users' to complete same function are different and the demands are also diversely. The rich cooperative relationships in VSCs provide a lot convenience for seeking service composing scheme and dynamic reconfiguration service composition is much easily realized by the combined action of CRC and VSC. Thus seeking service composing method can be transferred to the depth-first traversal algorithms or the broad-first traversal algorithms of graphic.

The companionship acts as the important parameters in the service composing and aids to seek the potential relationships among services. For example, a five-star hotel and the interest place of Forbidden City have two points in common, which is the both location is Beijing and both belong to travel domain. When user chooses to travel to Forbidden City, the VSC would recommend the five-star hotel.

IV. ANALYSIS OF SIMULATION EXPERIMENTS

This paper emphasizes on improving service organizing method, in order to make it achieved a higher degree of service clustering precision, get a more efficient service discovery and have a better auxiliary result of services composition. This experiments compared the recall ratio and the precision ratio of our model with OWL-S/UDDI service discovering method, contrast the different services composition time of the situation with and without VSC in order to test whether VSC could aid services composition or not. Two groups of tasks have been made in the paper. The experiments choose OWL-S describe web service, set the ontology and the service sample of OWLS-TC4, take the Hadoop computers group as the experimental environment, simulate the advantage of the centralized management and the spatial scalability under cloud environment.

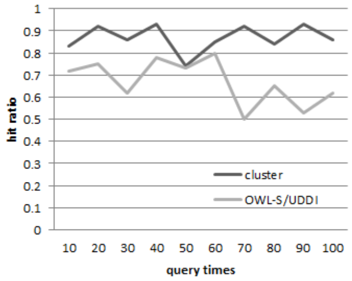

Figure 4. Precision ratio comparison

The first group is used to test the web service discovering results. The experimental data comes from the 1083 Web services of the OWLS-TC4 which involved communication, economy, education, food, medical treatment and tourism. By comparing the precision ratio from our service discovering method which is through service clustering to the OWL-S/UDDI Matchmaker in [16], results show cluster model has a better improvement on time. OWLS-TC4 supplies the experiment inquiry message. There are 6 inquiries for each domain. Existing services are divided into 10 times and added into service library one by one. Recording the eligible services under the two methods, the precision ratio is shown in Figure 4.

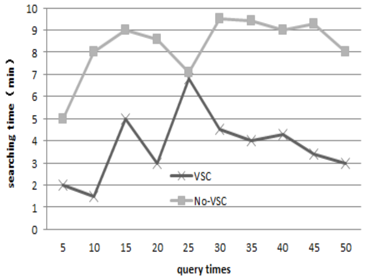

Figure 5. Service composing time comparison

The second group of experiments sends service requests repeatedly by simulating users'. The experiments only test and verify the effects putting relevance of ontology semantics while the effect of service composing has much relative to historical applying which the bigger historical data is, the higher the matching speed is. Using the same model, the effect of VSC has been tested and verified in each condition of whether the VSC exists or not under the composing method of QoS scheduling algorithm based on hybrid particle swarm.

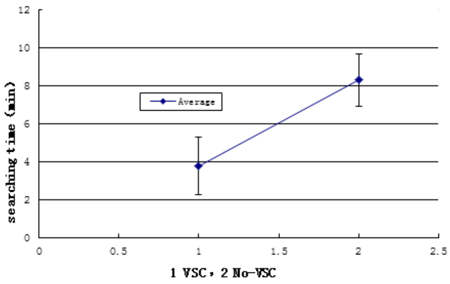

Figure 6. Graphicerrors of composing time

The precision ratio is increased obviously from Figure 4. It is evident that cloud clustering framework greatly raises the service matching precision as well as improves the searching efficiency. The framework does some flitting work while clustering so that it narrows the scope of service selecting and improves the precision in addition to the ontology similarity matching algorithm. Figure 5 and figure 6 shows that the VSC greatly reduce the time of service composing. But the nearly coincident point suggests that the search time is almost same for the services of all the quires have no cooperation relationship with and without VSC, and it also suggests that the degree of composing has much to do with query content. The two group of experiments show that the service organizing approach of cloud clustering and VSC has increased the service discovering efficiency, precision ratio and has a good auxiliary to service composition.

V. CONCLUSION

In order to reduce the pressure of registry as well as promote service discovery efficiency, this paper firstly makes a necessary improvement on semantics similarity algorithm based on ontology. And then, based on the new algorithm, the cloud clustering framework is built with a spinning tree form. Matching between two services or services and applying is accomplished by WSOSM. And the dynamic organizing of services and the personalized recommendation are achieved at the clouds by the clustering center. To make a better auxiliary effect of services composition, VSC sets the relevance services as a connected graph according to the cooperating relationship among services.

The experiments indicate that the new organization framework does raise the query efficiency, precision ratio and services composing efficiency. It is worthwhile to note that because of the imperfect of services ontology library and the high dependence of historical data and the designers, the recall, services recommendation and dynamic services reconstruction is still the subject of framework of the cloud clustering and cloud services community that needs to be explored in depth.

ACKNOWLEDGMENT

This paper is supported by the National Nature Science Foundation of China (No. 71171122) and the Shandong Provincial Natural Science Foundation of China (No.ZR2010FM003, 2009ERB019PF).

REFERENCES

[1] Elgazzar K, Hassan AE, Martin P. Clustering WSDL documents to bootstrap the discovery of Web services [C] // Proceedings of IEEE ICWS 2010, IEEE Computer Society, Editorial production by Lisa O'Conner, Miami, Florida, July 2010,pp.147-154.

[2] Sun Ping, Jiang Changjun. Using Service Clustering to Facilitate Process-Oriented Semantic Web Service Discovery [J]. Chinese Journal of Computers:Vol.31.2008.

[3] Liu XZ, Huang G, Mei H. Consumer-Centric service aggregation: Method and its supporting framework. Journal of Software: 2007,Vol.18(8), pp.1883-1895.

[4] Zeng L, Natallam B. QoS-aware Middleware for Web Service Composition[J].IEEE Transactions on Software Engineering: 2004,Vol.30(5):311-327.

[5] Liu Zhizhong, Wang Huaimin, Zhou Bin. A Two Layered P2P Model for Semantic Service Discovery[J]. Journal of Software: 2007, Vol.18 (8), pp.1922−1932.

[6] Duan Shoujian, Yang Chaofeng,Gan Jianhou. The Comprehensive Quantification Research of Semantic Similarity and Relevance Based on Domain Ontology [J]. Modern Library and Information Technology.2009 (11), pp.40-43.

[7] Christian Platzer, Florian Rosenberg, Schahram Dustdar. Web service clustering using multidimensional angles as proximity measures[J]. ACM Transactions on Internet Technology (TOIT) TOIT Homepagearchive:2009,Vol.9(3).

[8] Rada R, Mili H, Bicknell E, etal. Development and Application of a Metric on Semantic Nets[J]. IEEE Transactions on Systems, Man, and Cybernetics.1989, 19(1), pp.17-30

[9] Claire Fautsch, Jacques Savoy. Adapting the tf idf vector-space model to domain specific information retrieval[C]. Proceedings of the 2010 ACM Symposium on Applied Computing (SAC), 2010, pp.1708-1712.

[10] Hye-young Paik, Boualem Benatallah and Farouk Toumani3Towards Self-organizing Service Communities[J], IEEE Transactions on Systems, Man and Cybernetics. Part A: Systems and Humans: Vol.35, No.3.2005.

[11] Hu Chunhua, Wu Min, Liu Guoping, Xu Dezhi, An Approach to Constructing Web Service Workflow Based on Business Spanning Graph [J]Journal of Software: 2007,Vol.18(8), pp.1870−1882.

[12] Zhao Hongmei, Xia Yang, Zhao Qiang, Li Qi. Research on Chord-Based Semantic Web Service Discovery and Composition [J]. Microelectronics and computer:2010,Vol.27(8).

[13] Karl D Gottschalk, Stephen Graham, Heather Kreger, et al. Introduction to Web services architecture[J]. IBM Systems Journal(IBMSJ), 2008, 41(2), pp.170-177.

[14] Paolucci M, Kawamura T, Payne T R. Semantic matching of Web services capabilities[C]//Proc of the First Intl Semantic Web Conference, Italy: Sardinia, 2002, pp.333-347.

[15] Aggarwal R, Verma K, Miller J, Milnor W. Constraint driven Web Service composition in METEOR-S. In: Proceeding of the 2004 IEEE International Cnference on Services Computing, (SCC):2004, pp.23~30.

[16] Naveen Srinivasan, Massimo Paolucci, Katia Sycara. Adding OWL-S to UDDI, implementation and throughput. Robotics Institute, Carnegie Mellon University, USA. 2004. http://www.cs.cmu.edu/~softagents/papers/cr-swsws-paper.pdf.