Journal of Applied Sciences

ISSN 1812-5654

Journal of Applied Sciences 13 (13): 2579-2583, 2013
ISSN 1812-5654 / DOI: 10.3923/jas.2013.2579.2583

A Service Discover Fuzzy Matching Algorithm in Internet of Things

Shuren Zhu
Information Science School, Guangdong University of Finance and Economics,
Guangzhou 510320, Guangdong China

Abstract: Fuzzy matching algorithm of subscription is proposed in the optimization ideal here, it can increase the capability of system effectively that the algorithm filtrates unsuitability attributes to find the publish event quickly which matches the subscription basically. The foundation of the matching rules is discussed and designed, the structure principle of covering based filter and fuzzy matching based filter the structure of fuzzy vectors. The membership function of attributes and the threshold to judge whether the matching is success or not be defined. In order to support Web services based on quality of service, the broker mechanism for quality of service is introduced into the traditional Web service discovery model. On the basis of these, an algorithm about fuzzy matching is designed and experimental results were put forward.

Key words: Fuzzy matching, service discover, publish/subscribe, internet of thing

INTRODUCTION

Matching routing notification from producers to consumers are the main problems in large-scale publish/subscribe systems (Bornhovd *et al*., 2000; Crespo *et al*., 2000; Carzaniga *et al*., 2001; Fabret *et al*., 2001). Publish, in short, is some simple messages. It is called content. Clients can subscribe message publish event, or both (Fiege *et al*., 2002; Huang and Hector, 2002). Consumer subscribes message by issuing subscriptions. Essentially, these issuing subscriptions are information filters (Huang and Hector, 2001; Muhl, 2001). The customer may have many subscribe, when the consumer publishes a subscription, publish service will put forward the subscription to all publisher until the customer cancel the subscription (Muhl *et al*., 2000). Most distributed notification services use flooding, it's a waste of network resources (Opyrchal *et al*., 2000). In the filter, the simplest way is comparing event with all subscription information one by one. Some arithmetic try it's best to seek commonness between subscription information to reduce the comparison times in order to improve matching efficiency. Precision matching has high nicety, however, it has low efficiency and comparing each attribute is a complex thing, it can also results in bottleneck easily and long waiting-time for customer (Xue and Feng, 2004, 2005; Wang and Jin, 2005). While customer subscribes message, maybe the input of some attributes is not precision, if use precision matching, cannot find the suitable event.

The Internet of things technology architecture consists of four layers, namely the equipment perception layer, network layer, application layer, as well as security, privacy and trust layer. In the internet of things based on service-oriented architecture, two part of the service by the equipment service and the traditional Web service composition. With the traditional Web service.Than, equipment services have different characteristics. Equipment service is embedded in the physical device and can reflect the status of the physical world of real-time data services, while the traditional Web service is only encapsulates virtual entity business functions; equipment service deployment in resource constrained devices, but the traditional Web services deployed in the rich resources of the computer; the resources in the Internet of things limited mobility and wireless network oneself not reliability, equipment services are often in a highly dynamic environment, service often disappear or reappear, but Web service change traditional less frequent (Zhu and Deng, 2007; Wei *et al*., 2013).

The designation for filter based fuzzy matching was put forward and also discusses the construction of the fuzzy vector of attributers and the establishment of the matching rule. It is proposed membership function of the attribute and the threshold for checking success or not. Adopting fuzzy matching arithmetic to filtrate attributes can get the publish event that fit for subscription basically and improve the system capability availably. In order to support Web services based on quality of service, the broker mechanism for quality of service is introduced into the traditional Web service discovery model. On the basis of these, an algorithm about fuzzy matching is designed and experimental results were put forward.

FUZZY MATCHING ANALYSIS

Simple filter is consisted of a single predicate and compound filter is consisted of simple filter by combining them with Boolean operator. Any compound filter can be decompounded into standard form. Model filters as conjunctions of attribute filters that are simple filters and impose a constraint on the value of a single attribute. So, publishing a publish n, if n satisfies all attribute filter of F, n matches filter F. Moreover, a filter with an empty set of attribute filters matches any notification.

Defining the attribute filter as a tuple $AF_i = (n_i, Op_i, C_i)$, n_i is an attribute name, Op_i is a test operator and C_i is a set of constant that maybe empty. The name n_i determines to which attribute the constraint applies. If the notification doesn't contain attribute n_i, then AF_i returns 0 that means matching is failure, else Op_i returns value with the attributes and C_i. Supported that the type of operands are compatible with the used operator. The result of AF_i as the result of Op_i that returns a value between 0 and 1. For simply, use some simple notation {price = 20} instead of {(price, =, {20})}, for instance, a conjunctive filter consisting of three simple attribute filters is, {(name = "Computer science"), (price = 20), (publish = "Beijing")} By $L(AF_i) \subseteq dom(T_k)$ denote the set of all values that cause an attribute filter to match an attribute, for instance $\{V_i \mid Op_i(v_i, C_i) = a, a \in [0 \quad 1]\}$. To assume $L(AF_i) \neq 0$, if $n_1 = n_2 \wedge L(AF_1) \supseteq L(AF_2)$, the attribute filter AF_1 covers the attribute filter AF , $_2$denoted $AF_1 \supseteq AF_1$. For example, {price>10} covers {price>10}.

Proposition 1: Given two filter $F_1 = AF_1^1 \wedge ... \wedge AF_1^n$ and $F_2 = AF_2^1 \wedge ... \wedge AF_2^m$ that are conjunctions of attribute filters, the following holds:

- Assume: $\forall i \exists j, AF_i^1 \supseteq AF_j^2$
- Prove: $F_1 \supseteq F_2$

Proof: If an arbitrary notification n is matched by F_2 then n satisfies all AF^j_2, the fact together with the assumption implies that n also satisfies all AF^i_j. Therefore, n is matched F_1 too. So $F_1 \supseteq F_2$.

Proposition 2: Given two filters $F_1 = AF_1^1 \wedge ... \wedge AF_1^n$ and $F_2 = AF_2^1 \wedge ... \wedge AF_2^m$ that are conjunctions of attributes with at most one attribute filter for each attribute, the following holds: $F_1 \supseteq F_2$ implies $\forall i \exists j. AF_1^i \supseteq AF_2^j$.

- Assume: $\neg(\forall i \exists j. AF_1^i \supseteq AF_2^j)$
- Prove: $\neg(F_1 \supseteq F_2)$

Prove its contrary-denial proposition in order to prove the validity of the proposition.

Proof: Construct a notification n that matched F_2 but not matched F_1 to prove that F_1 does not cover F_2. The assumption implies that there is at least one AF_1^k that does not cover any Af^K_1. If there exists AF^1_2 that constrains the same attribute as such AF_1^k then choose for this attribute a value that matches AF^1_2 but not AF_1^k. Such a value exist because $L(AF_i) \neq 0$ and $AF_1^k \not\supseteq AF^1_2$. Add name/pairs for all other attributes that are constrained in F_2 so that they are mated by appropriate filter of F_2. The constructed notification matches F_2 but not F_1. So F_1 does not cover F_1.

Corollary 1: Given two filters $F_1 = AF_1^1 \wedge ... \wedge AF_1^n$ and $F_2 = AF_2^1 \wedge ... \wedge AF_2^n$ that are conjunctions of attribute filters with at most one attribute filter per attribute, $F_1 \supseteq F_2$ is equivalent to $\forall i \exists j, AF_1^i \supseteq AF_2^j$.

Proof: By Proposition 1 and 2.

ROUTING BASED ON COVERING

Supposing all brokers holds all subscription information, which can carry out identity-based routing table, but it is impossible in large-scale system, because it can generate overly large routing tables and frequent adjustment. Introducing similitude entry between filters can carry out minimizing routing entries in routing table. That needs to find the relationship between the matched notifications in the filters. For instance, if detecting two filters that match one notification, formally, the two filters F_1 and F_2 is equally.

Covering tests can reduce the number of routing entries and it can also reduce the number of control messages that must be forward. Supposing broker B has a neighbor broker H, broker B delivers a subscription/un-subscription S_2, if B has already forward a subscription S_1 to H and S_1 covers S_2, also broker B does not receive any corresponding un-subscription, so B does not need to forward S_2. If broker B receives a new subscription S from the neighbor broker U, broker B can drop those routing entries regarding U whose filter are covered by S. This implies that in case that B put forward un-subscription S to the neighbor H, all subscription that are covered by S of those routing entries regarding all brokers except H should be forward to H again.

Generally speaking, the routing based on subscription covering follows these rules below.

A subscription is not forwarded to a neighbor if a subscription that covers the former was forwarded to that neighbor that has not been canceled.

If a subscription is forwarded, the receiving broker deletes all routing entries whose subscriptions are covered by the new subscription and that refer to the same destination as new subscription.

A un-subscription is not forwarded to a neighbor if there is a subscription of a local client or another neighbor that covers the former.

If a un-subscription is forwarded to a neighbor, the sending broker also forwards a possibly empty subset of subscriptions covered by former.

It exists an exchange problem between network and the processor cost overhead, we can use statistical on-line adaptation of the filtering strategy to solve the problem. If the matching-rate is not up to or access to some threshold, then the broker which receives the information can request filtering again except the broker which is forward the information.

FUZZY MATCHING

Defining U as a domain, U maps to interval [0,1] ($U \rightarrow \mu_A(\mu)$), called the certain set A fuzzy subset of U (fuzzy set in short), written $A \in F(u)$. called μ_A eigenfunction of A. And called μ_A (u) membership function of A for U, written A(u).

After selecting correct membership function, to compute the matching degree through the function. The similitude degree between two fuzzy sets is matching degree. Use closing degree and Hamin distance to compute matching degree. Given two arbitrariness attribute sets A, B belonging to the same range, the Hamin distance between A, B is

$$H(A \cdot B) = \frac{1}{n}\sum_{i=1}^{n}\left|\mu_A(u_i) - \mu_B(u_i)\right| \quad (1)$$

So the matching degree between them is:

$$m = 1\text{-}H(A \cdot B) \quad (2)$$

Scale the approximately degree through the two formula above. The less distance, the more similar of the vectors, then the higher matching degree. When A = B, then H(A·B), m = 1 namely, that indicates two vectors superposition and homology completeness. To put forward a threshold α(α = 0.8 in this study), if $m \geq \alpha$, that means matching success, return true and output the successful publishing event, else that means matching failure, return false. Also put forward a sub-threshold β for each attribute (β = 0.5 in this study). The sub-threshold means that if some attribute membership function degree is 0, do not check the other attributes of the message matching or not in order to save time and improve efficiency:

$$T = \begin{cases} 1 & m \geq \alpha \\ 0 & m < \alpha \end{cases} \quad (3)$$

$$t = \begin{cases} 1 & A_i \geq \beta \\ 0 & A_i < \beta \end{cases} \quad (4)$$

Constructing fuzzy vector is the key of fuzzy matching, in the P/S system, different attribute has different impact in matching, checking each attribute is matching or not is a filtering process. Usually, need to check many attributes; so many attributes filters consist of conjunction filter. Suppose to publish/subscribe message, in this situation, most concerned attributes are defined, for example name, price and publish company and so on. Written {name, price, publish} as these variables. That is to say, this is a conjunction filter that consists of three attribute filters. Next, must choose a proper membership function. There are many typical membership functions, for example, trapezoid function, triangle function, rectangle function, bell function, S-function, Gauss function and so on. Select functions according as characteristic of attribute in practice.

After filtering the information by conjunction filter, the massage can be characterized into three types: Fully matching, uncertain matching and fully non-matching. Chooseing trapezoid function as membership function for the attribute name.

To choose triangle function as membership function for the attribute price, because customer usually cannot input absolute precision value.

For the attribute publish, simply speaking, we adopt rectangle function.

QoS PROXY MECHANISM

Introducing the QoS (Quality of Service) proxy mechanism in the SOA (Service Oriented Architecture) to support service publication and discovery of QoS description, ensure the demand of user to the Web service functional and non-functional attributes.

Web service provider: Responsible for the development and deployment of Web services is also defined service and QoS parameters and through the QoS agent release with QoS information Web service description.

Uddi registration center: Responsible for service registration and lookup, list helps Web service requesters search and order the required a variety of service types, description and location. The definition of a set of QoS classification tModel, CategoryBag data types used in the UDDI specification, in order to classify information keyName(description)/keyValue (category), the data type for each service defines a set of classification of tModel describes the Web service QoS to indicate QoS attribute information, to implement the expression of QoS information.

QoS proxy: Including QoS certification, QoS measurement and service selection functional modules. QoS authentication module to QoS demand for service providers QoS notice or request for quantifying the measurement module; QoS parameters was calculated for each service according to the method of the Qos value and the calculation results service selection module; service selection module function is based on demand of QoS service requestor, choose the most suitable to the needs of the service.

Web service requester: Is an application calls the Web service program, QoS Certification Center provided through the QoS proxy mechanism for service request QoS needs are quantified and then find the Web service through the Web service agent, call the required service, through the service provider to perform these services.

Assume the number of user services is k, the number of Qos demand is n in every service, Qos demand of user service is represented by Uqos (k, n) matrix.

where, S (i, j) = QoS property information; i = 1...k, service code; j = 1...n, Qos property information service code.

The weight of user service to Qos demand is represent by W (k, n) matrix, calculate weight matrix Q = UQoS×W'. Q (I, j) calculated by the Eq. 5:

$$Q\ (I, j) = UqoS\ (I, j) \times W\ (j, i) \qquad (5)$$

Measure of Qos service is described by WQos vector. calculated by the Eq. 6:

$$WQoS\ (I) = \Sigma Q\ (i, j)\ (j = 1...n) \qquad (6)$$

The most relevant services with the service request is the Web service with WQoS value.

```
ALGORITHM ( )

Input: subscription information S, publishing event set NP
Output: satisfied set NS in NP (α>0.8)
n = attribute number, α = 0.8, β = 0.5;
For (every publishing event N in publishing set NP)
    {
        μN (ui) =1, Sum = 0;
        For (every attribute ui in S)
            {
                If (μs (ui)<β)
                    Break;
                Else //compute Hamin distance between N and S.
                    Sum = sum +|μN(ui)-μS (ui)|;
            }// End for
        H(N·S) = Sum/n;
        m = 1-H(N·S);
        If (m≥α)
                Put N into NS set;
    }// End for
    END ALGORITHM
```

Fig. 1: Fuzzy matching Algorithm description

The fuzzy matching algorithm is shown in Fig. 1. Because the time complexity of the tree-based algorithm is superior to the counter-based algorithm, therefore, need only the algorithm is compared with the tree-based algorithm. Using C++ to realize fuzzy matching algorithm, the experimental environment is i5-2320/4G/2TB. The average execution time subscribe tree matching algorithm based on the need for 0.46 m sec. The average execution time of fuzzy matching subscription matching required for the 0.21 m sec. As can be seen, the fuzzy matching time than tree based matching algorithm of short time. This is because in the process of matching, threshold that many do not meet the requirements of the matching process terminated halfway.

CONCLUSION

The size of routing table will be reduced to use routing covering-based algorithm and then reduce the number of exchanged control information. The load efficiency evaluation of broker has been optimized by the fuzzy matching algorithm. It exists an exchange problem between network and the processor cost overhead, useing statistical on-line adaptation of the filtering strategy to solve the problem. If the matching-rate is not up to or access to some threshold, then the broker which receives the information can request filtering again except the broker which is forward the information. In the future work, the threshold α and sub-threshold β and the membership function will be optimized.

ACKNOWLEDGMENT

The study is supported by the Natural Science Foundation of Guangdong Province, China (Grant No.10151032001000001).

REFERENCES

Bornhovd, C., M. Cilia and C. Liebig, 2000. An infrastructure for meta-auctions. Proceedings of the 2nd International Workshop on Advance Issues of E-Commerce and Web-based Information Systems, January 8-9, 2000, Milpitas, CA., pp: 21-30-10.1109/WECWIS.2000.853855.

Carzaniga, A., D.S. Rosenblum and A.L. Wolf, 2001. Design and evaluation of a wide-area event notification service. ACM Trans. Compt. Syst., 19: 332-383.

Crespo, A., O. Buyukkokten and H. Garcia-Molina, 2000. E-client query subscription processing in a multicast environment. Data Eng., 18: 233-245.

Fabret, F., H.A. Jacobsen, F. Llirbat, J. Pereira, K.A. Ross and D. Shasha, 2001. Filtering algorithms and implementation for very fast publish/subscribe systems. Proceedings of the 2001 ACM SIGMOD International Conference on Management of Data, May 21-24, 2001, Santa Barbara, CA., USA., pp: 115-126.

Fiege, L., G. Muhl and F.C. Gartner, 2002. A modular approach to build structured event-based systems. Proceedings of the 2002 ACM Symposium on Applied Computing, March 10-14, 2002, Spain, pp: 385-392.

Huang, Y. and G.M. Hector, 2001. Replicated condition monitoring. Distributed Comput., 9: 15-26.

Huang, Y. and G.M. Hector, 2002. Exactly-once semantics in a replicated messaging system. Data Eng., 21: 134-148.

Muhl, G., L. Fiege and A. Buchmann, 2000. Evaluation of cooperation models for electronic business. Proceedings of the Conference of German Society for Computer Science on Information Systems for E-Commerce, November 2000, Berlin.

Muhl, G., 2001. Generic Constraints for Content-based Publish/Subscribe Systems. In: Cooperative Information Systems, Batini, C., F. Giunchiglia, P. Giorgini and M. Mecella (Eds.). Springer, USA., ISBN: 978-3-540-42524-3, pp: 211-225.

Opyrchal, L., M. Astley, J. Auerbach, G. Banavar, R. Strom and D. Sturman, 2000. Exploiting Ip Multicast in Content-based Publish-Subscribe Systems. In: Middleware, Sventek, J. and G. Coulson (Eds.). Springer-Verlag, USA., pp: 185-207.

Wang, J.L. and B.H. Jin, 2005. Data model and matching algorithm in ontology-based publish/subscribe system. J. Software, 16: 1625-1635.

Wei, Q., Z. Jin, G. Li and L. Li, 2013. Preliminary study of service discovery in internet of things: Feasibility and limitation of SOA. J. Frontiers Comput. Sci. Technol., 7: 97-113.

Xue, T. and B.Q. Feng, 2004. Design and implementation of content-based event notification service for digital library. Mini-Micro Syst., 25: 455-458.

Xue, T. and B.Q. Feng, 2005. Research on routing algorithm and self-configuration in content-based publish-subscribe system. J. Software, 16: 251-259.

Zhu, S.R. and T.T. Deng, 2007. Fuzzy matching routing filter in content-based publish/subscribe. Cent. South Univ. (Sci. Technol.), 38: 138-142.