2011 3rd International Conference on Signal Processing Systems (ICSPS 2011)
IPCSIT vol. 48 (2012) © (2012) IACSIT Press, Singapore
DOI: 10.7763/IPCSIT.2012.V48.30

Assessment Algorithm for Service Matching Based on Bloom Filter

Wendong Zhao, Jin Zhang, Laixian Peng, Dawei Niu and JingnanNie+

Institute of Communication Engineering, PLA University of Science and Technology
Nanjing 210007, China

Abstract. The main service search algorithms that used on content-based publish/subscribe system don't support service fuzzy matching effectively. This paper presents a service matching degree assessment method based on Bloom filter. Facilitated by this approach, an algorithm that supports service fuzzy matching has been proposed. The main idea of this algorithm is using Bloom filter to describe the service and request, and assessing the similitude degree of service and request by the similarity of Bloom filter vectors. Experimental and theoretical results show that this algorithm can support content-based service fuzzy matching by simple algebraic operations on Bloom filter. The evaluation accuracy rate is beyond 90%.

Keywords: Fuzzy matching; bloom filter; publish/subscribe.

1. Introduction

Content-based publish/subscribe[1] is a powerful paradigm for information dissemination from publishers to subscribers in large-scale distributed networks. A service specifies values of a set of attributes associated with it. Subscribers register their interests in service request through expressive subscriptions which specify complex filtering criteria by using a set of predicates over request attributes. Upon receiving a service description published by a publisher, the system matches the service to the request, which serve as filters, and delivers the service to the matched subscribers.

Fabret et al. proposed a content-based pub/sub scheme[2] defined as $U = \{A_1, A_2 ..., A_n\}$, where A_i means an atomic attribute. Each attribute consists of a *name*, *type* and *domain* and can be specified by a tuple < name, type, (min, max)>. Each service is a set of attributes that belong to set S and can be represented as $S = \{A_1 = c_1, A_2 = c_2, \ldots, A_n = c_n\}$, where c_i is the domain of A_i. Each request can be expressed as the logical operation on the attributes, for example, request Q can be described as $Q = \{(A_1 = v_1) \wedge (v_2 < A_2 < v_3)\}$. Without lost universality, the request is defined as $Q = \{A_1 = v_1, A_2 = v_2, \ldots, A_n = v_n\}$, where v_i means the restriction of A_i. If and only if $(\forall A_i \in Q \Rightarrow A_i \in S) \wedge (v_i \subseteq c_i)$, means request Q matches service S successfully.

Main matching algorithms used in content-based publish/subscribe system include tree-based algorithms[3-4], map-based algorithms[5] and XPath-based algorithms[6-7]. The main object of these algorithms is to decrease the operate times during the matching procession, but they all have the shortcoming described below:

They all are "Accurate Matching" algorithms, that is, if the service S matches request Q, every attribute that belongs to Q must belongs to S. But in the real world, users usually can not describe its need accurately. For example, if the user B's request is $Q = \{A_1 = v_1, A_2 = v_2, A_3 = v_3\}$, the service node A can give is $S = \{A_1 = v_1, A_2 = v_2, A_4 = v_4\}$. Maybe S is the service that B needs or S can partly meet Q, but the matching result is false.

For the algorithms can not support service fuzzy matching, according to the model Fabret has raised, this paper proposes a service matching degree assessment algorithm based on Bloom filter and based on this, we design an service fuzzy matching algorithms. This algorithm can also be used in service matching preprocess. That is, first, use this algorithm to filter almost all services that stored in local node with simple algebraic operations, then, use other algorithms to achieve accurate matching. Experimental and theoretical results show

+ Corresponding author.
E-mail address: nj_mouse@163.com.

that this algorithm can support content-based service fuzzy matching. The evaluation accuracy rate is beyond 90%.

The rest of the paper is organized as follows: section 2 gives an overview of related work. Section 3 shows basic definitions used in this paper, introduces the main idea and the algorithm. Section 4 discusses experimental setup and results. We conclude the paper in section 5.

2. Related Work on Bloom Filters

Standard Bloom filter[8] can be used to support membership queries because it uses a simple space-efficient data structure to represent a set. Bloom filters are widely used in network related applications, e.g., new network architecture design, route lookup, IP packet classification and network measurement.

The standard Bloom filter is a bit vector of *M* bits used to represent a set $S=\{s_1,s_2,s_3,\cdots s_n\}$ of *n* items. All bits in the vector are initially set to 0. Then, the Bloom filter uses *k* independent hash functions to map the set to bit vector. The domain of each hash function is *1* to *M*. For each item *s*, the bit $h_i(s)$ is set to *1*. To check whether an item *s* belongs to set *S*, we need to check whether all $h_i(s)$ are set to *1*. If not, *s* is not in the set *S*. If so, *s* is regarded as a member of *S* with a false positive probability, which suggests that set *S* contains an item *s* although it in fact does not.

Based on the standard Bloom filter, many other types of Bloom filter have been designed for specific applications, such as counting Bloom filter, compressed Bloom filter, split Bloom filter, dynamic Bloom filter.

The standard Bloom filter maps the items in the set to a vector. The vector must include some normal property about the set. Can we deduce the relation of different sets base on the Bloom filter vectors of these sets? In [9], the author designs three approaches for multi-attribute representation on network services based on parallel Bloom filter. But he does not talk about service matching based on this. In[10], the author first research the algebraic operations on Bloom filters from the point of set. In[11], the author analysis the effect on the vector when items in the set change and designs a quantitative assessment algorithm for the dynamic change of the set based on counting Bloom filter distance. Different from[11], this paper defines the similitude based on standard Bloom filter and uses the similitude to assessment the matching degree between the service and the request. Based on this, a fuzzy matching algorithm has been designed.

3. Algorithm and Analysis

3.1. Definition

In order to describe the algorithm clearly, we first show definitions as follows:

Definition 1 (Service description based on Bloom filter). For each attribute in Service description set, map the *name* and *type* of the attribute to the standard Bloom filter vector used *k* hash function. Denote as $BF^{k,m}(S)$.

Definition 2 (Coverage). The Coverage between service and request is defined as $C(S,Q)=\dfrac{|S\cap Q|}{|Q|}$. $|S\cap Q|$ means the total attributes with the same *name* and *type*.

Definition 3 (B-Coverage). The B-Coverage, based on Bloom filter, between service and request is defined as $C_{BF}^{k,m}(S,Q)=\dfrac{|BF^{k,m}(S)\cap BF^{k,m}(Q)|}{|BF^{k,m}(Q)|}$. $|BF^{k,m}(Q)|$ means the sum of the bits in the vector that are set to 1. $|BF^{k,m}(S)\cap BF^{k,m}(Q)|$ means the sum of the bits in these two vectors that the same bits are all set to 1.

For example, service *S* is $S=\{n_1,n_2,n_3,n_4\}$, it's bloom filter description is $BF^{k,m}(S)=[1,0,1,1,0,0,0,1,1,0]$. Request *Q* is $Q=\{n_1,n_2,n_3,n_5\}$, it's bloom filter description is $BF^{k,m}(Q)=[1,1,1,1,0,0,0,0,1,0]$. So

$$C(S,Q)=\frac{|S\cap Q|}{|Q|}=\frac{3}{4}.$$

$$C_{BF}^{k,m}(S,Q)=\frac{|BF^{k,m}(S)\cap BF^{k,m}(Q)|}{|BF^{k,m}(Q)|}=\frac{4}{5}.$$

3.2. Service matching alogorithm

The main idea of this algorithm is using Bloom filter to describe the service and request, and assessing the similitude degree of service and request by the similarity of Bloom filter vectors.

All services that the network can support use the same Bloom filter to generate the service description and are denoted as $S_{BF} = \{BF^{k,m}(s_1), BF^{k,m}(s_2), \cdots, BF^{k,m}(s_n)\}$. The request of the user also use the same Bloom filter to generate request description, denote as $BF^{k,m}(q_j)$. The coverage limitation is denoted as *Val*, *(0<Val<1)*. Then, the pseudo code of the fuzzy matching algorithm is shown in Fig. 1:

```
for ∀s_i ∈ S
    if C_BF^{k,m}(s_i,Q_j) ≥ Val
        put s_i into Set V
if |V|≥1
    return Set V
else
    return NULL
```

Fig. 1: Fuzzy mtching algorithm

When the node receives the request from the user, first compute the B-Coverage with each service saved in local node. If the B-Coverage satisfies the limitation, then, put the service to a set. When the computation is over, if the set is not empty, return the set, else return null.

Theorem 1 Based on the same Bloom filter and attribute universe, for the service *S* and the request *Q*, if $\bar{C}_{BF}^{k,m}(S_1,Q) > \bar{C}_{BF}^{k,m}(S_2,Q)$, then $\bar{C}(S_1,Q) > \bar{C}(S_2,Q)$. $\bar{X}()$ stands for the statistics average value.

Proof. Suppose the service *S*, denotes as $S = \{A_1 = c_1, A_2 = c_2, ..., A_n = c_n\}$, the request *Q*, denotes as $Q = \{A_1 = v_1, A_2 = v_2, ..., A_n = v_n\}$, and $|S| = n_1$, $|Q| = n_2$, $|S \cap Q| = n$, the vector length of the Bloom filter is *m*, the total number of hash functions is *k*. Depending on definition 1 to 3, we have:

The Coverage between *S* and *Q* is

$$C = \frac{n}{n_2} \tag{1}$$

We divide the map process into two steps. First, map the attributes belong to $S \cap Q$, then insert the attributes belong to $S - S \cap Q$ or $Q - S \cap Q$ into vector BF_S or BF_Q. The probability that $BF_s[i] \neq BF_q[i]$ is,

$$\begin{aligned}
&P\{BF_S[i] \neq BF_Q[i]\} \\
&= P\{BF_S[i]=0, BF_Q[i]=1\} + P\{BF_S[i]=1, BF_Q[i]=0\} \\
&= (1-\frac{1}{m})^{kn}(1-\frac{1}{m})^{k(n_1-n)}[1-(1-\frac{1}{m})^{k(n_2-n)}] \\
&+(1-\frac{1}{m})^{kn}(1-\frac{1}{m})^{k(n_2-n)}[1-(1-\frac{1}{m})^{k(n_1-n)}] \\
&= (1-\frac{1}{m})^{kn_1}(1-(1-\frac{1}{m})^{k(n_2-n)}) + (1-\frac{1}{m})^{kn_2}(1-(1-\frac{1}{m})^{k(n_1-n)}) \\
&= (1-\frac{1}{m})^{kn_1} + (1-\frac{1}{m})^{kn_2} - 2(1-\frac{1}{m})^{k(n_1+n_2-n)}
\end{aligned}$$

The probability that $BF_s[i] = BF_q[i] = 1$ is,

$$\begin{aligned}
&P\{BF_S[i] = BF_Q[i] = 1\} \\
&= 1 - P\{BF_S[i] \neq BF_Q[i]\} - P\{BF_S[i] = BF_Q[i] = 0\} \\
&= 1 - (1-\frac{1}{m})^{kn_1} - (1-\frac{1}{m})^{kn_2} + 2(1-\frac{1}{m})^{k(n_1+n_2-n)} \\
&-(1-\frac{1}{m})^{k(n_1+n_2-n)} \\
&= 1 - (1-\frac{1}{m})^{kn_1} - (1-\frac{1}{m})^{kn_2} + (1-\frac{1}{m})^{k(n_1+n_2-n)}
\end{aligned}$$

The probability that either $BF_s[i] = 1$ or $BF_q[i] = 1$ is,

$$P\{BF_S[i]=1 \vee BF_Q[i]=1\}$$
$$=1-P\{BF_S[i]=BF_Q[i]=0\}$$
$$=1-(1-\frac{1}{m})^{k(n_1+n_2-n)}$$

Then ,

$$\bar{C}_{BF}^{k,m}(S,Q)=\frac{|BF^{k,m}(S)\cap BF^{k,m}(Q)|}{|BF^{k,m}(Q)|}$$
$$=\frac{P\{BF_S[i]=BF_Q[i]=1\}}{P\{BF_Q[i]=1\}} \quad (2)$$
$$=\frac{1+(1-\frac{1}{m})^{k(n_1+n_2-n)}-(1-\frac{1}{m})^{kn_2}-(1-\frac{1}{m})^{kn_1}}{1-(1-\frac{1}{m})^{kn_2}}$$

$$(\bar{C}_{BF}^{k,m}(S,Q))'=\frac{-(1-\frac{1}{m})^{k(n_1+n_2-n)}\bullet\ln(1-\frac{1}{m})}{1-(1-\frac{1}{m})^{kn_2}}\geq 0 \quad (3)$$

Form equation 3, we can see, if we only consider the change of n, we can get

$$\bar{C}_{BF}^{k,m}(S_1,Q)>\bar{C}_{BF}^{k,m}(S_2,Q)\Rightarrow\bar{C}(S_1,Q)>\bar{C}(S_2,Q) \quad (4)$$

Depend on theorem 1 , we can drive the relation between $\bar{C}(S_1,Q)$ and $\bar{C}(S_2,Q)$ from the relation between $\bar{C}_{BF}^{k,m}(S_1,Q)$ and $\bar{C}_{BF}^{k,m}(S_2,Q)$. But in the real engineer, we use the relation between $C_{BF}^{k,m}(S_1,Q)$ and $C_{BF}^{k,m}(S_1,Q)$ to deduce the relation between $C(S_1,Q)$ and $C(S_1,Q)$. But for the false positive of the Bloom filter, the probability that $C(S_1,Q)>C(S_1,Q)$, but $C_{BF}^{k,m}(S_1,Q)<$ $C_{BF}^{k,m}(S_1,Q)$ is exist. In the next section, we conduct simulation to evaluate the accuracy of the algorithm.

4. Performance Evaluation

We use matlab to simulate and analysis the algorithm. This section focuses the simulation on three parts, (1) verify that the variation between the Coverage and B-Coverage is consistent by statistical method. (2) analysis the effect of parameters of the Bloom filter on our algorithm. (3) evaluate the accuracy of the assessment method.

4.1. Correctness analysis of the algorithm

The attributes sets of the service and request are created randomly depend on attribute universe set. Table 1 shows the simulate parameters.

TABLE I. PARAMETERS OF ALGORITHM SIMULATION

Name	Value
Attributes of universe	200
Attributes of each request	1~200
Attributes of each service	1~200
Hash functions	3
Length of the vector	1000
Simulate times	1000

We use the definition 2 to get the original Coverage, use the definition 3 to get the B-Coverage, use the formula 2 to get the theoretical average of the B-Coverage. The simulate result shows in Fig. 2 Each spot of the original Coverage and B-Coverage is the average of 1000 simulate results.

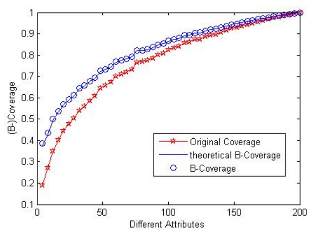

Fig. 2: Coverage simulation

From the figure 2 we can see that the statistical average of B-Coverage fully consistent with the theoretical average. At the same time, the variation of the B-Coverage is the same as the variation of the original Coverage. This means that we can deduce the matching degree between the original service and the original request by this assessing method.

We also can see from the figure 2 hat the larger the difference between the service and request, the larger the difference between the B-Coverage and original Coverage. This is because the false positive of the Bloom filter.

The figure 2 lso shows that for any spot it is always right that $\bar{C}_{BF}^{k,m}(S,Q) > \bar{C}(S,Q)$. This is because the original attribute of the Bloom filter. For any attribute *n*, we can deduce $\forall n \in S \cap Q \Rightarrow BF(n) \in BF(S) \wedge BF(n) \in BF(Q)$, but we can not deduce $n \in S \cap Q$ from $BF(n) \in BF(S) \wedge BF(n) \in BF(Q)$.

It should be noted that we use the relativity of the B-Coverage to deduce the relativity of the original Coverage. If the relativity unchanged, the false positive of the Bloom filter will do no effect on the assessment result.

4.2. Effect of Bloom filter parameters

Keep the service and the request unchanged, different hash functions and different vector length will generate different Bloom filter. Next, we will simulate the effect on the assessment result when the parameters of the Bloom filter changed. Table 2 shows the simulate parameters.

TABLE II. PARAMETERS OF SIMULATION

Name	Value
Attributes of universe	200
Attributes of each request	1~200
Attributes of each service	1~200
Hash functions	1~4
Step of the vector	500
Simulate times	1000

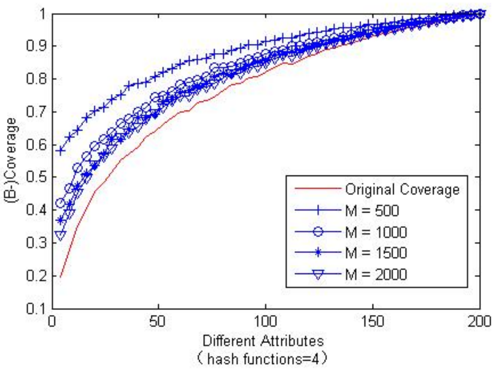

Fig. 3: Coverage comparison under different filter length

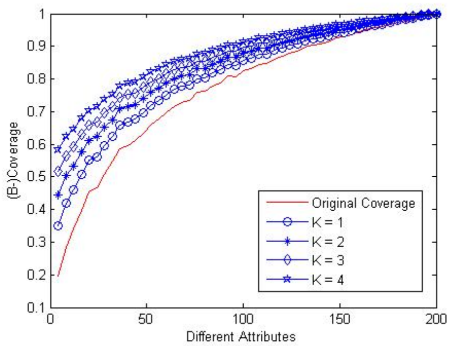

Fig. 4: Coverage comparison under differenthash function number

The simulate result shows in Fig. 3-4. Fig. 3 shows the changes with the number of hash functions growth and Fig. 4 shows the changes with the length of Bloom filter vector increases.

From the Fig. 3 and Fig. 4, we can see, under different simulate parameters the B-Coverage maintains the same variation with original Coverage. That is to say, from a statistical sense, given Bloom filter parameters, our algorithm would not make a mistake.

4.3. Analysis of the algorithm accuracy

Though theoretical analysis and simulation based on statistical shows that B-Coverage maintains the same variation with original Coverage, in real engineers, the case still exist the original Coverage between service *A* and request *B* is larger than service *B* and request *B* but the B-Coverage is on the contrary. This section simulates the accuracy of the algorithm. Table 3 shows the simulate parameters.

TABLE III. PARAMETERS OF APPLICATION SIMULATION

Name	Value
Attributes of universe	200
Attributes of each request	1~200
Attributes of each service	1~200
Hash functions	1~4
Step of the vector	500
Simulate times	1000

Table 4 shows the statistical result under different parameters.

TABLE IV. STATISTIC UNDER DIFFERENT PARAMETERS

Vector Length	Hash functions	Times	Error Times	Error Rate
500	1	200000	966	0.48%
500	2	200000	1089	0.54%
500	3	200000	1176	0.59%
500	4	200000	1471	0.74%
1000	1	200000	1096	0.55%
1000	2	200000	615	0.31%
1000	3	200000	792	0.40%
1000	4	200000	938	0.47%
1500	1	200000	567	0.28%
1500	2	200000	284	0.14%
1500	3	200000	260	0.13%
1500	4	200000	205	0.10%
2000	1	200000	312	0.16%
2000	2	200000	440	0.22%
2000	3	200000	317	0.16%
2000	4	200000	436	0.22%

The statistical results show that the error rate of the B-Coverage based assessment algorithm is super than 99%. It means that we can deduce the relativity of the original Coverage based on the relativity of the B-Coverage in the real engineers.

5. Conclusion

Based on the standard Bloom filter, we first define the Coverage and B-Coverage, then, analysis the relation between Coverage and B-Coverage. We conclude that from the statistical point of view, the B-Coverage maintains the same variation with original Coverage and we can deduce the matching degree between the original service and the original request by their description based on standard Bloom filter.

At last, we apply these concepts to content-based publish/subscribe system. We design a fuzzy matching algorithm. This algorithm can be used alone or be used as a preprocess algorithm with other accuracy matching algorithm to filter the invalid services and decrease the computation of second matching operation. Simulation shows the accuracy of this algorithm in real engineer is super than 90%. In the future work, we will try to apply this idea to overlay network based service discovery.

6. Acknowledgement

This work was supported in part under Grant BK2010103 from Nature Science Foundation of Jiangsu Province, China.

7. References

[1] Patrick TH. Eugster, Pascal A. Felber, Rachid Guerraoui, et al. The Many Faces of Publish/Subscribe[J]. Computing Surveys, 2003, 35: 114-131.

[2] F. Fabret, H.A. Jacobsen, F. Llirbat, et al. Filtering Algorithms and Implementation for Very Fast Publish/Subscribe Systems[C]. SIGMOD. ACM, 2001: 21-24.

[3] Gough KJ, Smith G. Efficient recognition of events in distributed systems[C]. Proceeding of the 18th Australasian Computer Science Conference. IEEE, 1995.

[4] Aguilera MK, Strom RE, Sturman DC, et al. Matching events in a content-based subscription system[C].Proceeding of Proceeding of the 18th ACM Symp. on Principles of Distributed Computing. Atlanta,1999.

[5] Campailla A, Chaki S, Clarke E, et al. Efficient filtering in publish-subscribe systems using binary decision diagrams[C].Proceeding of Proceeding of the ICSE 2001. Toronto,2001.

[6] Diao Y, Altinel M, Franklin MJ, et al. Path sharing and predicate evaluation for high-performance XML filtering[J]. ACM Transaction Database System, 2003, 28(4): 467-516.

[7] Peng F, Chawathe SS. XPath queries on streaming data[C].Proceeding of ACM SIGMOD. New York,2003.

[8] Bloom B. Space/time trade-offs in hash coding with allowable errors[J]. Communications of the ACM, 1970, 13: 422-426.

[9] Yu Hua Bin Xiao. Using Parallel Bloom Filters for Multiattribute Representation on Network Services[J]. IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, 2010.

[10] Kun Xie, Dafang Zhang, Jigang Wen, et al. Algebraic Operations on Bloom Filters[J]. Acta Electronica Sinica, 2008, 36(5): 86-874.

[11] Kun Xie, Jigang Wen, Dafang Zhang, et al. Quantitative Assessment Algorithm for the Dynamic Change of Data Set Based on Bloom Filter Distance[J]. Jounal of Chinese Computer Systems, 2009, 30(3): 411-416.

.