

Routing Technology of Publish/Subscribe System Based on Subscription Partition

Liuyang Jia *

School of Computer Science and Telecommunications
Engineering
Jiangsu University
Zhenjiang, China
E-mail:631907053@qq.com
*Corresponding Author

Zhiwen Zou

School of Computer Science and Telecommunications
Engineering
Jiangsu University
Zhenjiang, China
E-mail: zzw_yj@126.com

Abstract—A routing algorithm based on subscription partition of publish/subscribe is presented in this paper. This algorithm achieves the clustering partition of subscription by the common attribute of subscription predicates' distribution. Each agent manages different subset of the subscription. It greatly reduced the size of the routing table. In order to effectively reduce the number of forwarding events, events and subscriptions are forwarded by the unicast and multicast mixed method. Experiments showed that the algorithm proposed in this paper can effectively reduce the number of agents in the transmission process, and narrow the scope and number of matching operation acted in agent.

Keywords- publish / subscribe; subscription; matchin; routing; routing algorithm.

I. INTRODUCTION

There are two types of exact routing algorithm based on content routing algorithm. They are filter-based routing^[1,2], and multicast-based routing^[3,4]. For the filter-based routing, each jump will filter event's content and only forward the event to the destination address matched with the subscription in the event's forwarding process. However, in the forwarding path, because each node of agent needs to match the event before it is forwarded, this will bring a significant delay after the accumulation of multiple nodes.

The routing algorithm based on the partition of event/subscription space is a kind of multicast implementation. Its principle is dividing the logical space of the event/subscription into different areas. When an event is published, it will be mapped to the appropriate sub-region, and the agent responsible for the region, will complete the communication between publishers and subscribers in the region.

According to the different of clustering objects, partitioning technique can be divided into the event space partition(ESP) and subscription space partition(FSP). The main idea of the event space division method^[5,6] is mapping the event to a point in the event space. While the subscription is mapped to a region in the event space, and then the event space is divided into a number of disjointed event subspace. Each agent maintains a subspace. If the multidimensional space correspond by a subscription intersects with an event subspace, then this subscription is assigned to the agent

maintaining this space to manage. In forwarding process of event, the event needs to be mapped to the subspace which the event belongs to. Then it is forwarded to the appropriate subscribers by the agent maintaining the subspace. As the event subspaces are disjointed, the forwarding process of event can be seen as the unicast forwarding. However, if a subscription intersects with many event subspaces, then the agent correspond by these subspace needs to maintain the subscription. This increases the overhead of maintaining and updating the subscription.

Subscription space division method^[7] is mapping the subscription to a region in the multidimensional space, then divide the region into multiple subscription subspaces by some kind of aggregation method, each agent maintains a subspace. Therefore, the subscriptions agents maintain does not exist intersection. However, due to the point mapped by a event may fall into the intersection of multiple subspaces, so the event need to be forwarded to each agent corresponded by these subspaces, this increases the network overhead.

In the above two methods, they all need to map events and subscriptions to the multidimensional space, so the subspaces are partitioned by a higher complexity, while the clustering method proposed in this paper, simultaneously ensuring the routing efficiency, can make the division of the complexity be the constant level.

II. ROUTING ALGORITHM BASED ON PARTITION OF SUBSCRIPTION SET

A. Interval of interest

The event e is expressed as $\{A_1, A_2, \dots, A_k\}$, A_i represents a single attribute, $A_i = (\text{datatype}, \text{name}_i, \text{value}_i)$. Subscription Sub represents $\{P_1, P_2, \dots, P_n\}$, P_i represents a single predicate, $P_i = (\text{datatype}, \text{name}_i, \text{Op}_i, C_i)$. Given the attribute space, AttrNameTuple[N] = $(\text{name}_0, \text{name}_1, \dots, \text{name}_{N-1})$, defining the mapping $\Gamma : e \rightarrow \{0,1\}^N$, that is, event e is mapped to an N dimensional vector. $\{0,1\}^N$ is represented by a bit array, note as e_bit[N]. The mapping rules as follow:

$\forall A_i \in e$, if AttrNameTuple[j]=A_i.name_i, then e_bit[j]=1, and the remaining bits are zero. Similarly, subscription Sub can be mapped to $\{0,1\}^N$, this vector is expressed by a bit array Sub_bit[N]. Mapping rules as follows: $\forall P_i \in e$, if AttrNameTuple[j]=P_i.name_i, then Sub_bit[j]=1, the remaining bits are zero.

With binary tree index, subscription collection is clustering divided, this divides subscriptions with the same sub_bit into the same subset. In which, the layer i ($0 \leq i \leq N$) represent the i-th element name_i of AttrNameTuple, each leaf node $S^{\{0,1\}^N}$ represents a subset of subscription. In which, the sub_bit of subscription is equal with each other.

Obviously, each $S^{\{0,1\}^N}$ correspond with sequence $\{0,1\}^N$, and $\{0,1\}^N$ correspond with natural number n, so each $S^{\{0,1\}^N}$ can be uniquely expressed as a natural number. Since the partition tree's leaf nodes has at most $2N-1$, they are numbered from left to right by $1, 2, \dots, 2N-1$. Assume the number of event agents is M, then the number of subset assigned to each agent is $\frac{2^{N-1}}{M}$. Specify the proxy B_i to maintain the subscriptions included in the subset which number is between $\frac{2^N * (i-1)}{M} + 1$ and $\frac{2^N * i}{M}$. The interval agent maintained is known as interesting range, with i said. Make subs as the subscriptions set agent maintained.

Obviously, $\forall B_i, B_j, B_i.Subs \cap B_j.Subs = \emptyset$.

B. Routing Table Structure

Each agent has different interesting range. After the agent receives a subscription or event, the forwarding direction of the subscription and event is determined by the routing table.

Local subscription table (LSTB): it exists at the edge agent, it is used to record subscription ID and the customer's address sent by user who access to this agent. When the agent received event matching with the subscription in the table, it is used to forwarded event to the corresponding subscriber.

Interesting range table (IRTB): it exists at the edge agent, the table records the interval of interest allocated to each agent, its role is to determine a forwarding address of the event or subscription, after edge agent receives an event or subscription.

Interesting range subscription table (IRSTB): it exists in all agents, recording the subscription included in the subset the proxy's interesting interval corresponds. Its role is to query the source agent matching the subscription after the end of matching process.

C. Subscription routing algorithm

When the edge agent receives a subscription, it uses the mapping Γ to get the Sub_bit[N] of the subscription,

and then uses the Sub_bit to get the number of subset the subscription belongs to, the number is set to be k; Query IRTB and get the interval k belongs to, thereby getting the agent maintaining this interesting range. If the event agent find is itself, then the subscription is inserted into IRSTB; otherwise the subscription is sent to the target agent in same path, and the target agent insert the subscription into IRSTB.

```

function
Forword_Sub(Sub):
  Sub_bit[N]:=Γ(Sub);
  k:=conv(Sub_bit[N]);
  lookup this.IRTB, and find
  Bi, so that k ∈ Bi.I;
  if Bi= this
    this.LSTB ← LSTB
    ∪ (Sub, Sub.host);
    this.IRTB ← IRTB ∪ (Sub,
    this);
  else unicast(Sub, this) to
  Bi ;
  end Forword_Sub

```

Figure 1. Subscription forwarding algorithm

Duo to the uniqueness of target agent, subscription's forwarding can be seen as a one-to-one unicast process from a source agent to a destination agent. Assume that routers exchange each other and save the link information for the entire network, so as to equip the whole network topology. Thus they can independently calculate a route, finding a path from this agent to the destination agent. Pseudo-code of Subscription forwarding algorithm shown in Fig. 1.

D. Forwarding algorithm of Event

The subscription matching e is not necessarily all included in the subset of interesting range. When the edge proxy receives an event e and the subscriptions matched e scatter in many interesting range, we can look the agents maintaining these interesting range as a group, the edge proxy just need to sent e to all members of the group. In order to determine the communication object, we must firstly determine which IRSTB of agent contain the subscription interested in e, while the agents not contain the subscription matched e do not need to receive the event. The specific process is as follows:

Step1: get the e_bit[N] of event e by using mapping Γ .

Step2: search the partition tree according to e_bit[N]. If e_bit[i]=1, then execute branch parallel search, otherwise, only search the left subtree. Until come to the leaf node.

Step3: according to the number of subset searched, determining the range of interest each subset belongs to, viewing IRTB to get the target agent.

After determining the object of communication, the edge proxy will sent event e to the various target agents. The specific process is as follows:

- a) If the target agent is only one and the self agent, then directly perform matching operations in the local; otherwise, execute b).

- b) If the target agent is only one, not the self agent, then forward the event e to the target agent by unicast; otherwise, execute c).
- c) If the target agent is not only one and does not contain this agent, then forward the event e to each agent node by multicast; otherwise, execute d).
- d) If the target agent has multiple and contains the self agent, and the number of agent after deducting the self agent is one, then execute a), and use the forwarding strategy of b); otherwise, execute a and use the forwarding strategy of c).

After all target agents receive the event e, execute matching operation, and get the subscription matched event e, viewing IRSTB, finding the source agent of these subscriptions, then sent e to them. The specific process is as follows:

- 1) If the source agent is only one and is the self agent, then forward the event e to the appropriate subscriber according to LSTB, otherwise, execute 2).
- 2) If the source agent is only one, not the self agent, then forward the event e to the source agent by unicast; otherwise, execute 3).
- 3) If the source agent is not only one and does not contain this agent, then forward the event e to each source agent by multicast; otherwise, execute 4).
- 4) If the source agent has multiple and contains the self agent, and the number of agent after deducting the self agent is one, then use the forwarding strategy of 2); otherwise, use the forwarding strategy of 3).

III. SIMULATION AND PERFORMANCE

We simulated the routing algorithm proposed in this paper by NS2, and the experiment selected the following four evaluation indicators:

The average subscription forwarding hops (ASH):the number of agent nodes passed in the process where subscription was forwarded to the destination agent, ASH reflects the subscription routing efficiency.

The average event forwarding hops (AEH):the number of agent nodes passed in the process where events is sent to its interested agent, AEH reflect the efficiency of the event routing .

The average event forwarding delay (AED):the average time consumed in the process where events is sent to its interested agent, AED reflects the system's routing efficiency and scalability.

Average number of subscriptions (ANS):the average number of subscriptions maintained by agent. It is an important manifestation of the load balancing. The more self subscriptions distribute, the load of the event matching calculation is more balanced.

The size of property space in the experiment was set to be 10. The topology was a general graph structure. With the changing of agent's number (the total number of subscriptions is fixed at 30000), we compared routing algorithm based on the partition of subscription with reverse path forwarding algorithm^[8,9] in the 1,2,3 three indicators. The same time, with a fixed number of agents and subscriptions, and the subscription was average distributed, when we examine indicators 4, we randomly

selected a number of agents and compared the number of subscriptions they maintained, so as to measured ANS.

As to ASH, each agent sent a subscription in every 1 second, sending a total of 50 times, we counted the mean of ASH for subscriptions all proxy issued. Result was shown in Fig. 2. We can get some information from it. With the increasing of agent's number, ASH of the two algorithms increased, but the ASH of the partition routing algorithm was significantly less than the value obtained by the reverse path forwarding algorithm. This is because routing algorithm based on subscription division will forward the subscription to the appropriate agent by the best path. However, in the reverse path forwarding algorithm, each agent maintains a spanning tree whose root is itself, and the number of nodes in the spanning tree is usually much larger than the number of nodes on the optimal path. And when the number of tree is fixed, the average number of nodes in the spanning tree will increase as the total number of the agent increases.

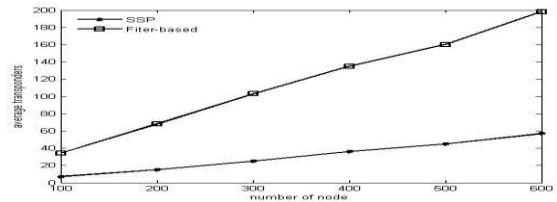


Figure 2. The ASH when the number of agents changes

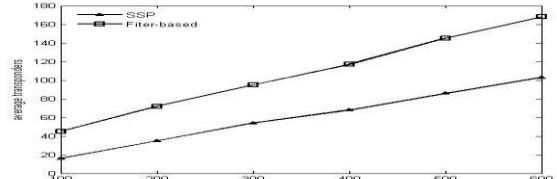


Figure 3. The AEH when the number of agents changes

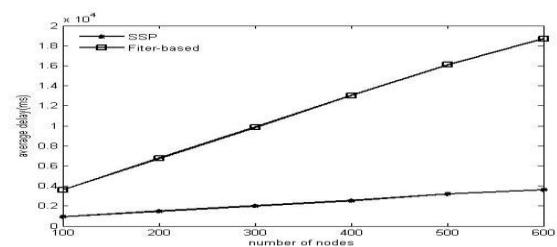


Figure 4. The AED when the number of agents changes

As to AEH, each agent sent a event in every 1 second, sending a total of 50 times. we counted the mean of AEH of all proxy. Result was shown in Fig. 3. It can be seen, with the increasing of agent's number, AEH of the two algorithms increased, but the AEH of the routing algorithm based on the division was less than the value obtained by the reverse path forwarding algorithm. This is because the algorithm proposed in this paper uses the tree path to forward events by multicast in the forwarding of events. However, in the reverse path forwarding algorithm, because subscription each agent maintains is different, it

is unable to determine the purpose agent of the event, needing to match and forward in the more agents.

As to AED, we separately investigated the performance of the two algorithms when agents' number is variable and the total number of subscriptions in system is variable. When the number of agents changed, each agent sent a event in every 1 seconds, sending a total of 50 times, we counted the average event forwarding delay. Result was shown in Fig. 4. With the increasing of agent's number, AED of the two algorithms increased. However, the AED of the division routing algorithm was significantly less than the value obtained by the reverse path forwarding algorithm. This is because the routing algorithm based on partition, in the event's forwarding process, not each agent receiving the event should do matching operation, but make the matching operation only limit to target agent. However, for the filter-based routing algorithm, each agent receiving an event need to do matching operation before the event is forwarded, this brings a lot of delay after the accumulation of all the nodes.

As to ANS, in the condition that subscriptions were average distributed, the number of proxy nodes was fixed at 100, each agent produced 100 subscriptions. After the system became stable, we randomly selected a few agents, sampling the number of their maintenance subscription.

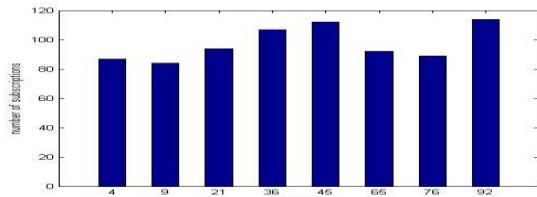


Figure 5. the results of Sampling the SSP

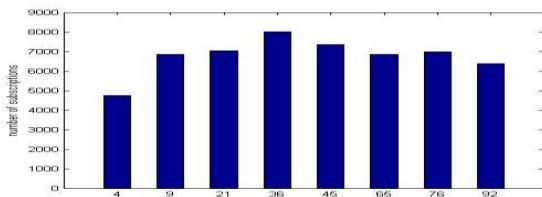


Figure 6. the sampling results of reverse path forwarding algorithm

The result was shown in Fig. 5, Fig. 6. As using the routing algorithm based on the division, the subscriptions' number of proxy was closed to the average. For a subscription, its maintaining agent is mapped by the distribution of the subscription's properties. Whereas, different agents have the different mapping property, so the subscriptions agents maintain don't exist intersection. At the same time, in the condition that subscriptions is evenly distributed, the subscriptions' number on the proxy

is roughly same. However, for the reverse path algorithm, a node may be located in multiple spanning trees, so the number of subscription such node maintain will exponentially increases.

IV. CONCLUSIONS

This paper analyzed the lack of filter-based routing algorithm and multicast-based routing algorithm. And designed a routing algorithm based on the partition of subscriptions' set. Describe the partition of Interesting interval. The realization method of the subscription routing algorithm and event forwarding algorithm is proposed. The basic idea of this algorithm is divided the subscriptions into multiple subsets. These subsets are maintained by each agent. Thus subscriptions maintained by each agent don't intersect. It can avoid the phenomenon that subscription is repeatedly maintained.

REFERENCES

- [1] S. Mikko, E.R. Christian, and A. Tuomas, et al, "Forwarding anomalies in Bloom filter-based multicast," IEEE INFOCOM, 2011, pp. 2399–2407.
- [2] J. Hemmes, M. Fisher, K. Hopkinson, "Predictive Routing in Mobile Ad-Hoc Networks," Next Generation Mobile Applications, Services and Technologies (NGMAST), 2011 5th International Conference on.117 - 122.
- [3] Xing Wang, Yuanquan Tan, Zhiheng Zhou , etal, "Reliable and energy-efficient multicast based on network coding for wireless sensor networks," Communications and Networking in China (CHINACOM), 2011 6th International ICST Conference on.17-19 Aug. 2011.1142 – 114.
- [4] Luo Li, Tuanfa Qin, Jianzhong Luo, etal, "A Routing Algorithm for Network Coding Multicast Based on Shareable Links," Dianxun Jishu / Telecommunications Engineering. Vol. 51, no. 3, Mar 2011, pp. 79-83.
- [5] I. Aekaterinidis, P. Triantafillou, "A scalable solution for DHT-independent content-based publish/subscribe data networks," Department of Computer Engineering and Informatics, University of Patras, Greece. Received 4 November 2009.
- [6] Sheng Gao, huili Guo, Pei Zhao, "Marshmallow: A Content-Based Publish-Subscribe System over Structured P2P Networks," Computational Intelligence and Security (CIS), 2011 Seventh International Conference , Dec. 2011, on.3-4.
- [7] Haifeng Liu, Zhaojun Wu, Milenko Petrovic, etal, "Optimized cluster-based filtering algorithm for graph metadata," 24 June 2010.
- [8] M.A. Tariq, B. Koldehofe, G.G. Koch, etal, "Distributed spectral cluster management: a method for building dynamic publish/subscribe systems," Proceedings of the 6th ACM International Conference on Distributed Event-Based Systems. ACM,2012
- [9] A. Carzaniga, M. Papalini, A.L.Wolf, "Content-based publish/subscribe networking and information-centric networking," Proceedings of the ACM SIGCOMM workshop on Information-centric networking
- [10] M.Guimaraes, L.Rodrigues, "A genetic algorithm for multicast mapping in publish/subscribe systems," Proceedings of the 2nd IEEE International Symposium on Network Computing and Applications. (2003) 67–7