International Journal of Advanced Computer Technology (IJACT)
ISSN:2319-7900

LARGE-SCALE PARTITION-BASED ROUTING FOR STRUCTURED P2P NETWORKS

Biao Dong, School of Computer & Software, Nanjing Institute of Industry Technology, Nanjing, China;
Jinhui Chen, School of Computer & Software, Nanjing University of Information Science & Technology, Nanjing, China

Abstract: A number of Distributed Hash Table(DHT)-based publish/subscribe(Pub/Sub) protocols have been proposed to address the issue of scalability in P2P networks. However, their routing state and control message overhead are enormous, the routing depth for notifications is unnecessarily long. We propose SGH, a large-scale partition-based overlay for P2P network, to provide the architecture of Pub/Sub routing. SGH relies on partitions where subscriptions and events are routed along distinct, albeit intersecting, sub-partitions. We design and implement the SGH event and subscription routing algorithms, and derive conclusions from the simulation experiments. The results show the simplicity of SGH model, and the scalability in Pub/Sub routing.

Introduction

Pub/Sub is an asynchronous communication paradigm that supports many-to-many interactions between a set of clients. A client can be an information publisher, an information subscriber, or both. Client interactions are data-centric: publishers describe their publishable events, subscribers express their interest in events, and Pub/Sub protocol delivers the published events to their corresponding event subscribers. The loose coupling of clients eliminates the burden of context information gathering and processing by resource constrained devices. Pub/Sub is different from the general unicast or multicast communication. When an event is published, Pub/Sub system doesn't specify a specific receiver. After each agent node receives an event, it decides which nodes should be propagated in the next step. Therefore, Pub/Sub system routing is also called content-based routing(CBR). Routing algorithm of Pub/Sub system resolves how to find an appropriate path in event broker networks, and how to efficiently and reliably route event to the relevant subscribers at a low cost. Network efficiency related to event forwarding is the most important design goals, which determines the size and scalability of the network.

P2P networking is a distributed application architecture that partitions tasks or work-loads between peers. Peers are equally privileged, equipotent participants in the application. We can classify networks as unstructured or structured. Unstructured P2P networks do not impose a particular structure on the overlay network by design, but rather are formed by nodes that randomly form connections to each other. In structured P2P networks the overlay is organized into a specific topology, and the protocol ensures that any node can efficiently search the network for a file/resource, even if the resource is extremely rare. Because of its advantages, such as decentralized control, fault tolerance and self-organization, P2P network has been widely recognized by the academia and industry. Many people use P2P technology to construct the Pub/Sub system to improve the ability to adapt to the changes of node failures and topology.

The rest of the paper is organized as follows. In Section 2, we review some related works. Our main methods including dominating set, scalable hierarchical graph, and event and subscription partitions in SHG are presented in Section 3. The event and subscription routing algorithms including subscription routing and event routing are given in Section 4, followed by the performance evaluations in Section 5. Finally, the paper is concluded in Section 6.

Related Work

DHT is a relevant area to our work. The DHT functionality supported by Pastry[1], Tapestry[2], Chord[3], and CAN[4] serves as a useful substrate for a range of large distributed systems; for example, Internet-scale facilities such as global file system, application-layer multicast, event notification, and chat services can all be layered over a DHT system. They employ the event broker whose node-Id is the hash of an event type (or a topic name) as the rendezvous node, or use P2P routing substrate for the event dissemination tree. Along with Tapestry, Castro presented Scribe, a scalable application-level multicast infrastructure. Scribe supports large numbers of groups, with a potentially large number of members per group[5].

Several content-based Pub/Sub systems based on DHT are proposed. Gupta provided Meghdoot, which adapts content-based Pus/Sub systems to DHT based P2P networks in order to provide scalable content delivery mechanisms while maintaining the decoupling between the publishers and the subscribers[6]. Triantafillou leveraged the advantages of the Chord DHT to build a content-based publish-subscribe system, and provided solutions supporting efficiently subscriptions with range predicates in Chord-based Pub/Sub systems[7]. Muthusamy proposed algorithms for supporting content-based Pub/Sub in which subscriptions can specify a range of interest and publications a range of values. The algorithms are built over a DHT abstraction and are completely decentralized. Load balance is addressed by subscription delegation and a bottom up tree search technique[8].

There are other aims as to content-based P2P Pub/Sub systems. Setty presented PolderCast which aims to achieve relay-free, fast and robust dissemination over a scalable overlay with a minimal maintenance cost[9]. Nakayama discussed the causal dependency of application events handled by subscription, publication, and notification semantics[10]. Kyoungho An presented the content-based filtering discovery protocol, which eliminates unnecessary discovery messages for participants according to matching topic names and endpoint types[11]. Akiyama proposed a Pub/Sub environment that can be used with software defined networks[12]. Ahmed presented peer-assisted Pub/Sub which is a hybrid broker P2P content-based Pub/Sub system with varying event sizes[13]. Bellavista focused on the ability of Pub/Sub infrastructures to offer cost-effective, scalable, and quality-aware data distribution in emerging wide-scale and highly dynamic communication environments[14].

Recent works in content-based Pub/Sub systems based on P2P networks exploit DHT to support CBR. Some of these approaches require the existence of a so-called rendezvous node where subscriptions meet events, thus easily creating bottlenecks. Still, their routing state and control message overhead are enormous. In conclusion, a distributed Pub/Sub needs a structured overlay network, but it must be designed with great care since the underlying P2P architecture has a significant effect on the performance. In this paper, we propose a new P2P overlay, called SHG, suitable for large-scale Pub/Sub. The design guidelines are as follows.

(1) A partition-based structure rather than a tree is preferred for a reliable and adaptive event and subscription dissemination tree. Subscriptions and events are routed on distinct, albeit intersecting, SHG partitions.

(2) The concept of dominating node(DN) is proposed. The choice of DN isn't fixed, the probability that node is selected as DN is equal. According to the size of the network and the Pub/Sub situation, the ranges of nodes, which DNs are responsible for, are dynamically determined.

(3) SHG can be used not only for flexible topic or type based systems by nature, but also as a routing substrate for highly selective content-based systems.

Our Methods

A. Dominating Set

Dominating set(DS) is defined as a node set in graph G, which makes the other nodes in the graph adjacent to a certain node in the node set. A node in DS is called DN. Otherwise, the node is called non-dominating node(non-DN). DS is an important concept in partition-based routing. It has been widely used in the layout problem of Pub/Sub nodes. In CBR, each node tends to publish events and to propagate subscriptions. In order to reduce traffic congestion in networks and improve performance of network, typically, nodes are divided into partitions. For any node in a partition, it publishes or subscribes only through its nearest node in DS. Therefore, it is required to reasonably set up the locations of these DNs.

We assume that the graph G=(V, E) is a connected graph, and S is a subset of V. If any node in V-S is adjacent to a certain node in S, then S is called as DS of G, and denoted as DS(G). Any node in V- DS(G) is called as non-DN, and denoted as non-DS(G). Let t be a given positive integer, for any node y in V-S, there is a node x, the distance between x and y is less than t, then S is called as G's distance t-dominating set, and denoted as t-DS(G). In t-DS(G), the minimum number of nodes is called distance t-dominating number. A t-DS(G) is perfect if and only if the t-DS(G) satisfies the following two conditions: first, the distance between any two nodes in t-DS(G) is not less than t; second, the distance between any node in any t-DS(G) and any node in any non-t-DS(G) is less than t. We mainly discuss t-DS(G) problem, so as to ensure that the distance between any node in a DS(G) and any node in non-DS(G) isn't greater than a constant.

In this section, we propose a CBR method based on DS, shown in Fig.1, to describe the model of CBR management layer integrated Pub/Sub mechanism and DS.

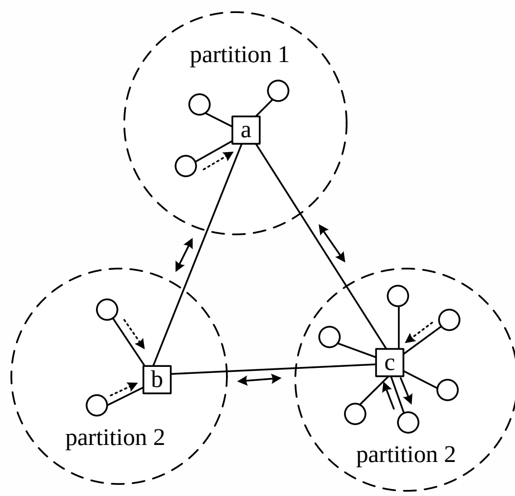

Figure 1. A CBR Method based on DNs

In the legends of the paper, we use rectangle and circle to represent DN and non-DN respectively. Dashed arrow and solid arrow indicate propagating subscriptions and publishing events respectively.

In Fig.1, a graph is divided into three partitions, which are responsible for publishing/subscription by DNs a, b and c respectively. In each partition, a non-DN is only connected to its neighboring DN, and DNs are connected to form a network. So, the graph is divided into two levels. In such networks, A DN adjacent to a non-DN is responsible for publishing/subscription of the non-DN. A DN publishes events to other DNs, and maintains a subscription table for all adjacent non-DNs.

B. Scalable Hierarchical Graph

SHG is an approach for modeling and implementing partition-based routing for P2P networks. The concepts associated with SHG are defined as follows.

Definition 1 identifier(ID). Let $\chi=\{\sigma_i|-1\leq i\leq n\}$ be an alphabet, a non-empty finite set. An ID over χ is a finite sequence of elements from χ, and satisfies one of two ways.

(1) σ_{-1} is an ID.

(2) $\chi^* = \chi^0 \cup \chi^1 \cup \cdots$ is an ID set, for $\left\{\sigma_{i_1}, \sigma_{i_2}, \cdots, \sigma_{i_p}\right\} \subseteq \chi - \{\sigma_{-1}\}$, where $p(p \geq 1)$ is integer number, and $i_1<i_2<...<i_p$.

Some notations concerning the ID are as follows. First, let $\chi^* = \chi^0 \cup \chi^1 \cup \cdots$, where $\chi^0=\{\sigma_{-1}\}$, $\chi^1=\chi-\{\sigma_{-1}\}$, $\chi^i=\{\mathrm{ID}(i)|\ i>1\}$. Second, given an arbitrary ID, its alphabet is denoted by χ(ID). Its length is the number of elements in ID, and is denoted |ID|.

Definition 2 addition & subtraction. Given two arbitrary ID_1 and ID_2. Addition is signified by the circle sign(∘), $ID_1{\circ}ID_2 = \sigma_{i_1}\sigma_{i_2}\cdots\sigma_{i_p}$, where $\left\{\sigma_{i_1}, \sigma_{i_2}, \cdots, \sigma_{i_p}\right\} = \chi\{ID_1\} \cup \chi\{ID_2\}$, and $\sigma_{i_1}\sigma_{i_2}\cdots\sigma_{i_p}$ is an ID. Subtraction is signified by the minus sign(-). ID_1-ID_2 represents the operation of removing all characters which are not in χ-$\chi(ID_2)$.

Definition 3 adjacent node. Given two arbitrary non-DNs ID_1 and ID_2, they are called the adjacent IDs iif $ID_1 - ID_2 = \pm 1$. Given two arbitrary DNs ID_3 and ID_4, they are called the adjacent IDs iif $|ID_3| - |ID_4| = \pm 1$.

Definition 4 SHG. A SHG over χ, denoted by $\mathrm{SHG}_{|\chi|}$(V,E) (also called $\mathrm{SHG}_{|\chi|}$ for short), is an undirected graph with V=χ^*, and then $(ID_1, ID_2) \in$ E iif ID_1 and ID_2 are adjacent nodes.

C. Event and Subscription Partitions in SHG

A CBR protocol must guarantee that the routes followed by published events and by subscriptions intersect in at least one node. We generalize this notion by assuming that the dissemination of events and subscriptions is performed through different partitions of the nodes. Subscriptions are disseminated through partitions S_i they cover without overlapping the whole set of system nodes. Similar definitions hold for the event partitions ε. The two types of partitions are related by the following constraint: each event partition must intersect all subscription partitions, and vice versa. When these properties holds, an event sent to all the nodes in a partition ε is received by at least one node in each subscription partition S_j. In the paper, we enforce this topological property in SHG.

Definition 5 subscription partition. Given $\chi=\{\sigma_i|-1\leq i\leq n\}$ and $\chi_{\geq k}=\{\sigma_i|i\geq k$, k is an arbitrary integer $1<k<n\}$. Subscription partition $S_i = \{ID|ID \in a_i{\circ}(\chi - \chi_{\geq k})^*, a_i \in \chi_{\geq k}\}$ denotes the set of nodes. In S_i, a node may propagate its subscription along the entire subscription partition the node belongs to.

Definition 6 event partition. Event partition $\varepsilon = \{ID|ID \in \chi_{\geq k}{}^*\}$ denotes the set of nodes. In ε, a node may propagate its events along the entire event partition.

Definition 7 DNs in SHG. Given a subscription partition $S_i = \{ID|ID \in a_i{\circ}(\chi - \chi_{\geq k})^*, a_i \in \chi_{\geq k}\}$, the two nodes, a_i and $a_i{\circ}\sigma_0\sigma_1\cdots\sigma_{k-1}$ are called DNs of S_i.

By the definitions 5-7, the event partition of SHG is formed by just all DNs in SHG. Each subscription partition contains two DNs. t-DS(SHG) is equal to $\left\lfloor k/2 \right\rfloor$. According to the distance between non-DN and two DNs in the same subscription partition, each subscription partition is divided into two sub-partitions.

Fig.2 illustrates partition-based routing on SGH_5 with 32 nodes, where $\chi=\{\sigma_{-1}, \sigma_0, \sigma_1, \sigma_2, \sigma_3, \sigma_4\}$, n=5, and k=3. SGH_5 includes four subscription partitions, they are SGH_3, $\sigma_3{\circ}\mathrm{SGH}_3$, $\sigma_4{\circ}\mathrm{SGH}_3$, $\sigma_3{\circ}\sigma_4{\circ}\mathrm{SGH}_3$, respectively. SGH_5 includes one event partition $\varepsilon=\{\sigma_{-1}, \sigma_0\sigma_1\sigma_2, \sigma_3, \sigma_0\sigma_1\sigma_2\sigma_3, \sigma_4, \sigma_0\sigma_1\sigma_2\sigma_4, \sigma_3\sigma_4, \sigma_0\sigma_1\sigma_2\sigma_3\sigma_4\}$. The nodes with the same label constitute one sub-partition. There are 8 sub-partitions in Fig.2. When unambiguous, we drop the IDs for readability. The top-right portion of the figure represents subscription partition SGH_3, DNs σ-1, $\sigma_0\sigma_1\sigma_2$ denote two DNs, namely Pub/Sub agents. Solid arrows show event routing generated by the non-DN publishers, and dashed arrows indicate subscription propagating generated by the non-DN subscribers. As can be seen here, subscriptions are restricted to a single subscription partition, instead of the whole system.

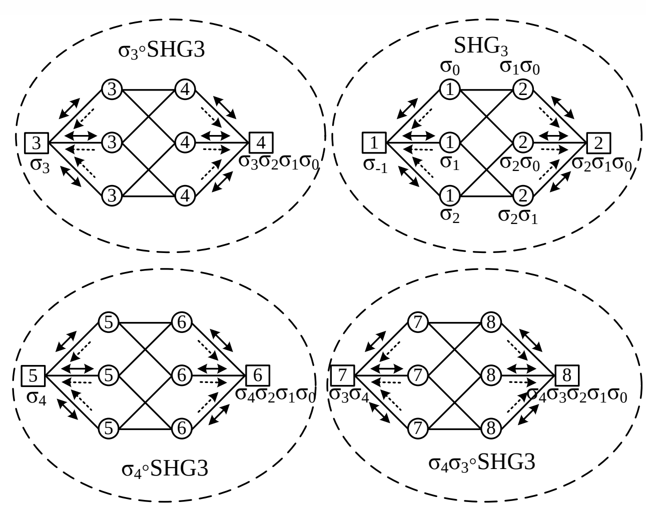

Figure 2. Partition-based routing on SHG_5

ISSN:2319-7900

Event and Subscription Routing

A. Subscription Routing in SHG

When a subscription to a new pattern *p* is issued by a DN (e.g., σ_0 in Fig.2), the DN don't propagate the subscription p along the entire subscription partition SHG_3, and only creates an item for p in its subscription table. If a second subscription p is issued by another node on the same subscription partition (e.g., σ_1, $\sigma_0\sigma_1$), these non-DNs propagate p only up to the closest DNs(σ_{-1}, $\sigma_2\sigma_1\sigma_0$ in our case). However, if this second subscription is issued on another sub-partition different from the first one, the subscription propagates independently throughout that second sub-partition. A subscription q for a different pattern is propagated on the same method. Essentially, the sub-partition is used as a sort of overlay spanning the subscription partition. The subscription for the new pattern is first routed to a DN of the corresponding sub-partition, leveraging the routes already in place. From there, the subscription is then used to reach all the nodes in the sub-partition, which insert the new pattern in their routing tables. This scheme ensures that all the patterns are disseminated in the same way in each sub-partition, and no loops are created.

The key to subscription routing is to establish the corresponding routing tables in the corresponding sub-partitions. The algorithm for DNs to construct subscription routing tables is described as follows.

void procedure sub_table_constructing (identifier ID, integer k, subscription p) {
    alphabet low_s, high_s, diff_low_s, diff_high_s;
    identifier high_ID, low_ID;
    integer diff1, diff2;
    low_s=χ(ID)-χ($\sigma_n\sigma_{n-1}\ldots\sigma_k$);
    high_s=χ(ID)-χ($\sigma_{k-1}\sigma_{k-2}\ldots\sigma_0$);
    for $(\text{high\_ID} = \{\}; \forall\sigma_i \in \text{high\_s}; \text{high\_s} - \{\sigma_i\})$ do
        high_ID=high_ID$\circ\sigma_i$;
    for $\left(\text{low\_ID} = \{\}; \forall\sigma_j \in \text{low\_s}; \text{low\_s} - \{\sigma_i\}\right)$ do
        low_ID=low_ID$\circ\sigma_j$;
    diff1=|ID-high_ID|;
    diff2=|ID-high_ID$\circ\sigma_{k-1}\sigma_{k-2}\ldots\sigma_0$|;
    if (diff1>diff2) {
      //Select high_ID as DN, and create a subscription route between high_ID and ID.
        for $(\text{diff\_low\_s} = \chi(\text{ID} - \text{high\_ID}); \forall \text{x} \in \text{diff\_low\_s}; \text{diff\_low\_s} - \{\text{x}\})$ do
        //If there is a subscription p in the subscription table of node ID∘x, then modify the p item.
        //Otherwise, create a new subscription p in the node's subscription table.
        if (p ∈ sub_table(ID°x))
            Modify p item of the subscription table of node ID∘x;
        else
            Create a new subscription item p in the subscription table of node ID∘x;
      }
    else {
        //Select high_ID$\circ\sigma_{k-1}\sigma_{k-2}\ldots\sigma_0$ as DN, and create a subscription route between high_ID$\circ\sigma_{k-1}\sigma_{k-2}\ldots\sigma_0$ and ID.
        for ($\text{diff\_high\_s} = \chi(\text{ID} - \text{high\_ID}^\circ\sigma_{k-1}\sigma_{k-2}\cdots\sigma_0)$; $\forall \text{x} \in \text{diff\_low\_s}; \text{diff\_high\_s}^\circ\{\text{x}\}$) do
          if (p ∈ sub_table(ID°x)) then
            Modify p item of the subscription table of node ID∘x;
          else
            Create a new subscription item p in the subscription table of node ID∘x;
    }
}

B. Event Routing in SHG

As for events, our routing strategy is made of two constituents. First, events are routed through their own sub- partitions. In this respect, event propagation behaves exactly like subscription propagation. In Fig.2, an event published by non-DN σ_0 propagates along the entire sub-partition rooted at DN σ_{-1}. Second, An event disseminated along an event partition crosses all the subscription partitions containing matching subscriptions, as long as it is routed along the whole event partition. When an event hits a subscription partition containing matching subscriptions, it is captured by that subscription partition and duplicated along it, and by following the path established by subscriptions. Fig.3 illustrates event partition and event propagation on SHG_5. Events are propagated to all subscription partitions by corresponding DNs, instead of to all non-DNs directly.

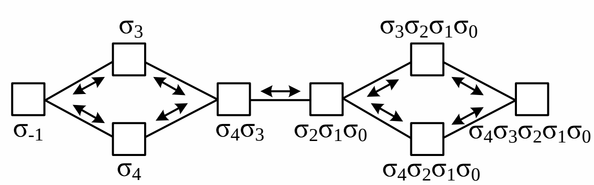

Figure 3. Event partition and event propagation on SHG5

Interestingly, SHG provides alternatives for system deployment, by changing the size of subscription and event partitions. For instance, it's likely to minimize the traffic caused by events, at some additional cost for routing subscriptions. However, the inverse choice may be beneficial in a system where subscriptions outnumber events. Mainstream CBR approaches do not provide this flexibility, as they route events and subscriptions on the same overlay, typically a tree.

The key to event routing is to establish routing between adjacent DNs in the event partition. The algorithm for event routing is described as follows.

```
void procedure event_routing(identifier DN_ID, event e) {
    subscription p;
    identifier_set e_partition_pre;
    alphabet high_s;
    identifier high_ID;
    //If event e matches item p in the subscription table of
    source DN DN_ID, then e is propagated within the sub-
    partition corresponding to the node.
      if (∃p ∈ subscription_set ∧ match(p, e))
        broadcast(e, sub-partition(DN_ID));
    //Broadcast event e between these DNs which are adja-
    cent to DN_ID in event partition
    for (e_partition_pre = χ*(σ_n σ_{n-1} ⋯ σ_k); ∀x
                ∈ e_partition_pre; e_partition_pre
                − {x}) do
         broadcast(e, sub-partition(x◦DN_ID));
    //Get another DN, which is in the same subscription par-
    tition with DN_ID.
    high_s=χ(ID)-χ(σ_{k-1}σ_{k-2}…σ_0);
    for (high_ID = {}; ∀σ_i ∈ high_s; high_s − {σ_i}) do
      high_ID=high_ID◦σ_i;
    another_DN_ID=(high_ID<>DN_ID)?          high_ID:
    high_ID◦σ_{k-1}σ_{k-2}…σ_0;
    //Broadcast event e between these DNs which are adja-
    cent to another_DN_ID in event partition
    for (e_partition_pre = χ(σ_n σ_{n-1} ⋯ σ_k); ∀x
                ∈ e_partition_pre; e_partition_pre
                − {x}) do
         broadcast(e, sub-partition(x◦ another_DN_ID));
    //Get the other DNs, which are adjacent to DN_ID, and
    satisfying the condition ||DN_ID|-|other_DN_ID||=1.
      DN_length =|DN_ID|;
    others = {x|x ∈ χ* ∧ (|x| = DN_length + 1 ∨ |x|
                = DN_length − 1};
    for (; ∀x ∈ others; others − {x}) do
     broadcast(e, sub-partition(x));
  }
```

Simulation Experiment Validation And Evalution

We validated SHG by simulation. The simulator uses PeerSim, a discrete event simulator for large-scale distributed systems. Results are averaged over 100 simulation runs with different seeds.

In SGH, any node can be used as a publisher and(/or) subscriber. Traffic analysis depends on the characteristic parameters of Pub/Sub system. Let N_p and N_e respectively represent the number of different subscriptions and events in a system in the given period of time. If a subscription p obeys uniform distribution, $\sigma(p)=k_p$, where $0\le k_p\le 1$ is a constant. Let $\alpha(p)$ be the probability of a node of being a subscriber for a pattern p. Let $\beta(e)$ be the probability of a node to be a receiver for event e. We compared against Scribe, and define BT as the ratio of the traffic generated by SHG to the traffic generated by Scribe. We set some parameters: N_p=100, N_e=30*N_p, α=0.65%, β=15%, $|\chi|$=6, and k=$|\chi|$/2. This is an extreme scenario particularly suitable for subscription forwarding.

Fig.4 shows the results of our $BT(e)$ comparisons for increasing values of the number of nodes from 64($|\chi|$=6) to 1024($|\chi|$=10).

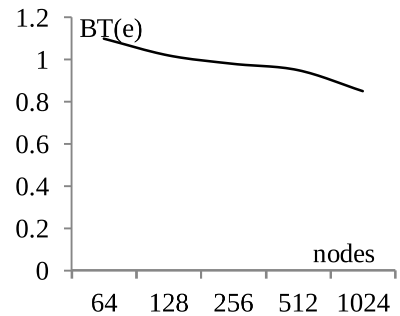

Figure 4. Traffic difference comparing SHG event routing and Scribe

Fig.5 shows the results of our $BT(p)$ comparisons for increasing values of the number of nodes from 64($|\chi|$=6) to 1024($|\chi|$=10).

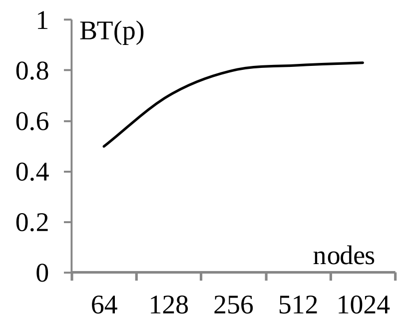

Figure 5. Traffic difference comparing SHG subscription routing and Scribe

Fig.4 and Fig.5 shows the flexibility of SHG that blends well with many different workloads. Similarly, SHG exhibits very good scalability because event propagation involves each time a different set of nodes according to publisher.

Conclusions

In this paper, we propose SGH, a large-scale partition-based Pub/Sub for P2P networks, to provide the architecture of Pub/Sub routing. SGH is based on event and subscription partition model. We design and implement SGH event routing and subscription routing algorithms, and derive conclusions from the simulation experiments. The results show the simplicity of SGH model, and the scalability in Pub/Sub.

ISSN:2319-7900

Acknowledgments

This work was sponsored by Qing Lan project(Jiangsu province, china) and open fund project of Jiangsu provincial research and development center of intelligent sensor network engineering technology, china (ZK13-02-03, Software technology, platform and application for sensor network).

References

[1] A. Rowstron, and P. Druschel, "Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems," Middleware 2001, Springer Berlin Heidelberg, pp.329-350, 2001.

[2] B. Y. Zhao, J. Kubiatowicz, and A. D. Joseph, "Tapestry: An infrastructure for fault-tolerant wide-area location and routing," University of California, Berkeley, California, report No. UCB/CSD-01-1141, 2001.

[3] I. Stoica, R. Morris, D. Karger, et al, "Chord: A scalable peer-to-peer lookup service for internet applications," ACM SIGCOMM Computer Communication Review, Vol. 31, No. 4, pp. 149-160, 2001.

[4] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker, "A scalable content-addressable network," ACM, Vol. 31, No. 4, pp. 161-172, 2001.

[5] M.Castro, P. Druschel, A. M. Kermarrec, et al, "SCRIBE: A large-scale and decentralized application-level multicast infrastructure," Selected Areas in Communications, IEEE Journal, Vol. 20, No. 8, pp. 1489-1499, 2002.

[6] A. Gupta, O. D. Sahin, D. Agrawal, et al, "Meghdoot: content-based publish/subscribe over P2P networks," Proceedings of the 5th ACM/IFIP/USENIX international conference on middleware, Springer-Verlag New York, Inc., pp. 254-273, 2004.

[7] P. Triantafillou, and I. Aekaterinidis, "Content-based publish-subscribe over structured P2P networks," Proceedings of the 4th International Workshop on Distributed Event-Based Systems, pp. 104-109,2004.

[8] V. Muthusamy, and H. A. Jacobsen, "Infrastructure-free content-based publish/subscribe," IEEE/ACM Transactions on Networking, Vol. 22, No. 5, pp. 1516-1530, 2014.

[9] V. Setty, M. Van Steen, R. Vitenberg, et al, "PolderCast: fast, robust, and scalable architecture for P2P topic-based pub/sub," Proceedings of the 13th international middleware conference, Springer-Verlag New York, Inc., pp. 271-291, 2012.

[10] H. Nakayama, D. Duolikun, T. Enokido, et al, "A P2P Model of Publish/Subscribe Systems," Broadband and Wireless Computing, Communication and Applications (BWCCA), 2014 Ninth International Conference on IEEE, pp. 383-388, 2014.

[11] K. An, A. Gokhale, D. Schmidt, et al, "Content-based filtering discovery protocol (CFDP): scalable and efficient OMG DDS discovery protocol," Proceedings of the 8th ACM International Conference on Distributed Event-Based Systems, ACM, pp. 130-141, 2014.

[12] T. Akiyama, Y. Kawai, K. Iida, et al, "Proposal for a New Generation SDN-Aware Pub/Sub Environment," ICN 2014, pp. 210-214, 2014.

[13] N. Ahmed, M. Linderman, and J. Bryant, "PAPaS: peer-assisted publish and subscribe," International Journal of Business Process Integration and Management, Vol. 7, No. 2, pp. 128-136, 2014.

[14] P. Bellavista, A. Corradi, and A. Reale, "Quality of Service in Wide Scale Publish-Subscribe Systems," Communications Surveys & Tutorials, IEEE, Vol. 16, No. 3, pp. 1591-1616, 2014.

Biographies

BIAO DONG received his BS in Computer Application Technology from Nanjing University, China (2001). He received his MS and PhD in Computer Science and Technology from the Hohai University, China (2004), Nanjing University of Science and Technology, China (2011), respectively. Currently, he is an associate professor in the School of Computer and Software at Nanjing Institute of Industry Technology, Nanjing, China. His research interests include new services for wireless sensor and actuator networks, service provisioning and management, advanced software architectures and component-based distributed applications design.

JINHUI CHEN received her BS in Computer Science and Technology, MS in System Analysis and Integration from Nanjing University of Information Science & Technology, China, in 1992 and 2002, respectively. Currently, she is an associate professor and supervisor of graduate students in the School of Information and Control, Nanjing University of Information Science & Technology. Her research interests include wireless networks, mobile computing and approximation algorithm design and analysis.