Wuhan University Journal of Natural Sciences

2012, Vol.17 No.1, 031-035

Article ID 1007-1202(2012)01-0031-05

DOI 10.1007/s11859-012-0800-0

TSPN: Term-Based Semantic Peer-to-Peer Networks

□ GAO Guoqiang[1,2], LI Ruixuan, LU Zhengding[1]

1. School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan 430074, Hubei, China;

2. School of Media and Communication, Wuhan Textile University, Wuhan 430073, Hubei, China

Abstract: In this paper, we propose Term-based Semantic Peer-to-Peer Networks (TSPN) to achieve semantic search. For each peer, TSPN builds a full text index of its documents. Through the analysis of resources, TSPN obtains series of terms, and distributes these terms into the network. Thus, TSPN can use query terms to locate appropriate peers to perform semantic search. Moreover, unlike the traditional structured P2P networks, TSPN uses the terms, not the peers, as the logical nodes of DHT. This can withstand the impact of network churn. The experimental results show that TSPN has better performance compared with the existing P2P semantic searching algorithms.

Key words: Term-based Semantic Peer-to-Peer Networks (TSPN); peer-to-peer; semantic parsing; semantic DHT

CLC number: TP 393

0 Introduction

With the development of computer and network technology, the computer model has undergone dramatic changes. As a new network service model, Peer-to-Peer (P2P) networks are changing our way of life. P2P refers to a class of systems and applications that employ distributed resources to perform a critical function in a decentralized manner. Semantic P2P is the semantic overlay network (SON) which consists of peers linked by the semantic in P2P networks. Semantic P2P uses inherent semantic information of resources to construct and manage P2P networks, which can improve query efficiency of the network. A semantic P2P network is also a complex network, and it has some characteristics of the complex network, such as small world and power-law. We can utilize these features to improve semantic P2P networks.

The network topology directly affects search performance, and semantic search is still subject to this rule. Therefore, to improve the performance of semantic search, we must improve the semantic topology of P2P networks. In this paper, we focus on structured semantic P2P networks. Structured P2P networks use Distributed Hash Table (DHT) to distribute resources. In order to achieve semantic search in DHT, general approaches are to parse the semantic description of resources into terms and distribute these terms into DHT. Although this strategy can achieve efficient full-text search, the maintenance cost is very high, especially for highly dynamic P2P networks.

P2P semantic search is based on the resource contents, which attracts great interests from both industry

Received date: 2011-03-03

Foundation item: Supported by the National Natural Science Foundation of China(60873225, 60773191, 70771043), National High Technology Research and Development Program of China (2007AA01Z403), Wuhan Youth Science and Technology Chenguang Program (200950431171)

Biography: GAO Guoqiang, Ph. D., Lecturer of Wuhan Textile University, research direction: peer-to-peer computing and information retrieval. E-mail: ggq@ smail.hust. edu.cn

and academia [1-3]. In SemreX [4], a peer always chooses high similarity peers as neighbors when it joins the network. This strategy can cluster the peers with similar semantic in unstructured P2P networks. Many studies use ontology as a semantic analysis tool. Jason uses this method in Ref.[5], which proposes a query transformation method to efficiently collect as many relevant resources from the distributed information systems as possible. In pSearch [6], documents in the network are organized around their vector representations (based on modern document ranking algorithms) such that the search space for a given query is organized around related documents, achieving both efficiency and accuracy. However, pSearch has a poor performance in network churn. Disconnection failures arise when semantic P2P (SP2P) systems that use adaptive query routing methods treat temporary mapping faults as permanent mapping faults. Mawlood-Yunis A R *et al* [7] propose the fault tolerant adaptive query routing (FTAQR) algorithm to resolve the problem.

In this paper, we propose a new architecture TSPN (Term-based Semantic P2P Networks) to distribute resources and peers into DHT, which can effectively withstand the impact caused by dynamic P2P networks. A node of traditional DHT is a peer of the P2P network. The resources of the network are managed by different DHT nodes according to their DHT addresses. If a peer leaves the network, DHT must be reconstructed. Therefore, the overhead of traditional DHT is very large in dynamic environments. In TSPN, we take the term as the node of DHT, and each peer is managed by the term. Since a term may belong to multiple peers, thus the structure of DHT will not be affected by the offline of peers. We use Chord [8] as DHT topology in our approaches. To reduce the size of DHT, we use TF-IDF (Term Frequency-Inverse Document Frequency) mechanism to extract only those terms that best represent the resources of peers as the nodes of DHT. Each peer constructs full semantic index for its resources, and queries are directed to similar semantic peers to search through TSPN. In order to achieve semantic search, we also use the synonym strategy to cover more peers in DHT routing.

1 TSPN

TSPN is an efficient P2P architecture which archives semantic search. In this section, we will discuss three key parts of TSPN in detail: semantic parsing, DHT model, and query mechanism.

1.1 Semantic Parsing

For semantic search, the resources owned by peers must be organized in the semantic structure. We assume that the resources of peers are documents, so the smallest unit of semantic analysis is the term. As for other resources, such as pictures, movies, etc., we can use the description of these resources to carry out semantic parsing. In TSPN, for each peer, we build a full text index of its documents to achieve the semantic search. When a query is forwarded to a peer, TSPN returns relevant results to users through querying its full-text index. In order to include more semantic similar results, we not only search query keywords from the index, but also search the synonyms of query keywords.

In TSPN, we use DHT to locate peers, and publish the terms rather than the resources for semantic search. To reduce overhead, we only release those terms that can best represent the peers. We use the TF-IDF to compute the importance of a term for a peer. TF (Term Frequency) usually refers to the number of occurrences of a term in a document. In this paper, TF refers to the number of occurrences of a term in a peer. IDF (Inverse Document Frequency) is a measure of general importance of a term. More documents a term belongs to, the smaller the IDF of the term, and vice versa. Therefore, a term with high frequency in a peer and with low frequency in the entire documents collection can generate high TF-IDF weights. We extract only those terms whose TF-IDF is greater than a certain threshold form a peer, and publish them to our DHT network.

1.2 Semantic DHT

The biggest problem of structured P2P is that it must reconstruct the entire DHT when a peer leaves the network. In a dynamic P2P network, its DHT is always changing, which will bring huge maintenance overhead. In TSPN, we propose a new DHT structure to alleviate this problem. We use Chord [8] as the infrastructure, and the term is used as the node of Chord's DHT. DHT topology of TSPN is shown in Fig. 1. Unlike traditional DHT, we use terms in peers of the network as logical nodes of DHT, rather than peers, and these peers are managed by the terms. As can be seen in Fig. 1, the example network includes seven peers and three terms are extracted from the resources in these peers to represent them. We take terms t_1, t_2, t_8 as the nodes of DHT, and a term manages the peers that include it. For example, the peers' p_1, p_3, p_5 which own the term t_1 are managed by the term t_1. Since a term may be included in multiple peers, peer's departure will not change the distribution of nodes of TSPN's DHT but simply change the contents of the nodes of DHT. Therefore, TSPN can effectively resist network churn.

Structured P2P networks commonly use bootstrap server to guide a new peer joins. In TSPN, we assign a primary peer and a secondary peer for each term. The bootstrap server in TSPN stores part of (term, primary peer) of the network. The secondary peer will replace the primary peer once the primary peer is failure, and the new primary peer is also responsible for selecting a new secondary peer. In the primary peer and the secondary peer, they record all peers managed by the term. When a peer p joins the network, it first extracts a series terms from its documents. Then, the peer lets these terms to join DHT of TSPN, respectively. For a term t_i, the peer p queries the bootstrap server through the hash value of t_i, and it will obtain a term t_j, wherein hash address is close to its term t_i. By querying the primary peer of t_j can obtain the final primary peer of t_i. If the primary peer of t_i does not exist, TSPN will generate a new node t_i, and take the peer p as the primary peer of t_i. To balance the load, a peer can transfer its primary role to other peers in the same group for a term. The route table of peers in the network are shown in Table 1. For normal peers, they only record the primary/secondary peers corresponding to their terms. For a term, its primary/secondary peers not only record all peers in the same group which are sorted according to their TF-IDF weights, but also record the primary peer of its next hop term. To simplify, we just list a next hop term in Table 1. Although the physical route is still to rely on peers, DHT needs to be reconstructed only when both the primary peer and the secondary peer are off-line at the same time. As a result, TSPN can ensure a good stability compared to the traditional DHT.

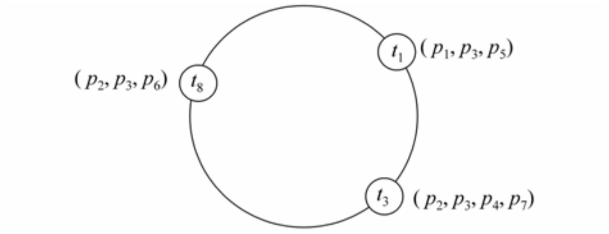
t_1 (p_1, p_3, p_5)
(p_2, p_3, p_6) t_8
t_3 (p_2, p_3, p_4, p_7)

Fig. 1 DHT topology of TSPN

Table 1 Route table of primary peer, secondary peer, and normal peer for Fig. 1

Primary/secondary peer				Normal peer		
Term	Hash value	All peers	Next hop	Term	Hash value	Primary/secondary peer
t_1	1	p_1,p_3,p_5	p_2	t_1	1	p_1
t_3	3	p_2,p_3,p_4,p_7	p_3	t_3	3	p_3
t_8	8	p_2,p_3,p_6	p_1	t_8	8	p_2

1.3 Query Mechanism

To start a query, TSPN first parses the query words to form a serial of query terms. Second, for a query term t, TSPN query the primary peer of the term whose hash value is close to that of t to obtain the next hop primary peer, and so on, until locating the primary peer p_t corresponding to the term t. In order to cover more semantic, TSPN not only searches t but also queries the synonyms of t. Finally, TSPN obtains the peers corresponding to the query terms from the primary peers, and searches the full text indexes of these peers. The search results are returned to the initiating peer, and are shown to the users after merging and sorting.

Since a term may manage a large number of peers, searching the full-text indexes of all of these peers will bring significant overhead. To reduce overhead, TSPN only selects a certain number of peers to query after locating a query term. In this paper, we use two different approaches to select which peers to query: Optimal and Random. Optimal selects these peers whose TF-IDF weights are higher to query their full text indexes. TF-IDF weight indicates the correlation between a peer and a term. Therefore, this strategy is more likely to find relevant documents. However, Optimal always queries the peers whose TF-IDF weights are high, which could lead to that some peers will not be covered in query. Random randomly selects a certain number of peers to query, which can effectively distribute queries. This method is also beneficial to the discovery of new resources.

2 Performance Evaluation

In this section, we first simply introduce our experiment methodology, and then we analyze the efficiency of TSPN.

2.1 Simulation Methodology

To save time and increase efficiency, we use PeerSim as the simulation-driven kernel. PeerSim has been designed to cope with P2P networks. It can support simulations on a very dynamic environment. It has great scalability and can support thousands of peers. In our experiments, we implement Chord protocol in Peersim.

In order to reveal the real environment, the object popularity in our experiments follows a Zipf-like distribution. This distribution states that some personal content is highly popular and the rest has more or less the same low popularity. In equation (1), the Zipf-like probability mass function [9] is provided, where C denotes the number of personal content items and α is the exponent characterizing the distribution, $x \in \{1, \cdots, C\}$.

$$P_{\text{Zipf-like}}(x) = \frac{x^{-\alpha}}{\sum_{j=1}^{C} j^{-\alpha}} \tag{1}$$

In Ref.[10] Backx *et al* show, with a number of practical experiments using popular P2P file sharing applications, that α is usually between 0.6 and 0.8. In our experiments, we take α as 0.7 because our system follows that rule as well. To assign terms to each of peers in the network, we generate 5000 distinct terms and assign each term a frequency according to equation (1).

2.2 Performance in Churn

Churn arises from continued and rapid arrival and failure (or departure) of a large number of peers in P2P networks. We evaluated the efficiency of TSPN in structured P2P networks in churn. Experiment results verified the resilience of TSPN in churn. In the simulation, node join and voluntary departure are modeled by a Poisson process with a mean rate of R, which ranges from 0.05 to 0.5. A rate of R = 0.05 corresponds to one node joining and leaving every 20 s on the average. Figure 2 plots the average query hit ratio versus the node join/leave rate. As we can see, query hit ratio of TSPN is 90% at R = 0.5, however, Chord only has 55%. Because TSPN uses terms as the logical nodes of DHT, DHT of TSPN has small influence at network churn. However, Chord must reconstruct its DHT when a peer leaves the network, which not only brings a huge maintenance costs, but also affects the search efficiency.

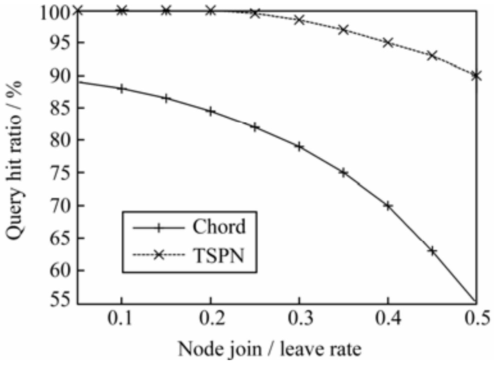

Fig. 2 Performance of different network architectures in network Churn

2.3 Precision

Precision is the proportion of relevant documents searched to all documents searched. Precision is an important evaluation factor, especially for semantic search. In our experiments, we use TF-IDF strategy to calculate the relevance of the query words and the documents searched, and thus distinguish whether a document is relevant or not. For comparative analysis, we use our experimental environment to simulate pSearch and SemreX. The pSearch uses traditional structured networks to store resources, and each node is responsible for managing a number of terms obtained from the documents. However, SemreX uses unstructured networks to distribute resources, and calculate similarity to achieve semantic search during random walk. In the experiments, we run 20 separate queries each algorithm and calculate their precision.

Figure 3 shows experimental results for three different semantic search strategies in precision. As we can see, TSPN has the best performance and reaches at an average of 89% precision. pSearch has the worst effect, and only 78% on the average. This is because TSPN uses the terms with high TD-IDF weights to represent the documents, and uses the query terms to match the document terms in searching. The documents searched are usually related documents after merge and sorting, unless the query documents are very little or not in the network.

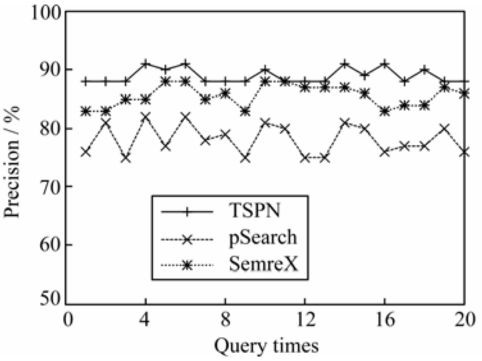

Fig. 3 Precision for 20 queries for TSPN, pSearch, and SemreX

2.4 Query Performance

In this set of experiments, we measure query mechanism of TSPN. Because a term in TSPN may manage many peers, especially for those popular terms, searching all related peers will lead to huge overhead. To reduce these overhead, TSPN only selects a certain number of peers to query. In TSPN, we use two select mechanisms: optimal and random. First, we use precision metrics to evaluate the two select mechanisms. All the configurations are the same as previous experiments. As can we see in Fig. 4, the precision of Optimal is higher than that of Random. On the average, optimal improves 6% compared to Random in precision. Optimal always selects the most relevant peers to query, so it has better performance than Random in precision.

We also design another experiment to evaluate the two peer select algorithms in query mechanism of TSPN. We launched 20 similar semantic queries, and then observe the cumulative number of peers searched. Figure 5 shows experimental results for optimal and random. Random uses 405 peers to search for 20 similar queries, while optimal only uses 167. Because the 20 queries are similar, a lot of query keywords are located in the same peers. Therefore, the mechanism of optimal makes TSPN have fewer cumulative peers to search, as compared to random. Although optimal can get a higher precision, Random can find more new content. In some applications, returning a more extensive content to uses may be a good idea. As a result, system designers decide to use Optimal or Random according to the actual situation.

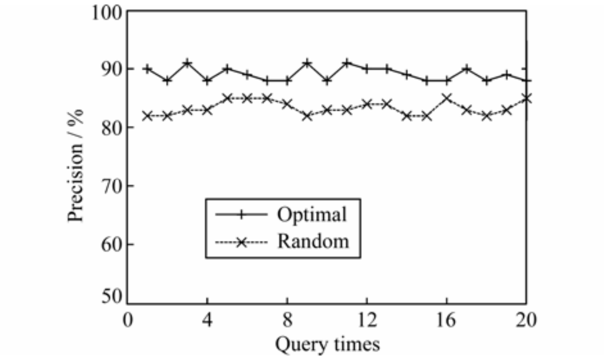

Fig. 4 Precision for 20 queries for optimal and random

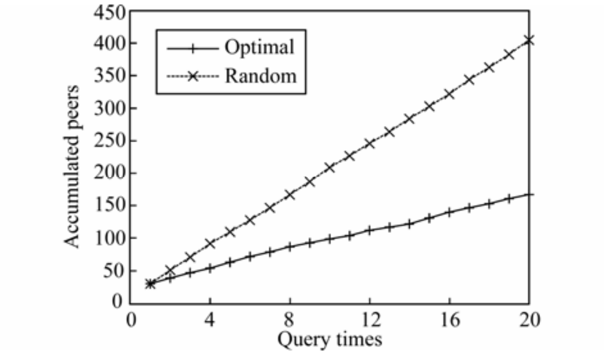

Fig. 5 Accumulated peers for 20 queries for optimal and random

3 Conclusion

In this paper, we propose a novel semantic searching algorithm for the structured P2P networks, named TSPN. Comparing with the existing P2P semantic searching algorithms, TSPN has better performance. The experimental results show that TSPN is more able to withstand the impact of network churn, and it has higher precision compared with other algorithms. In future work, we will further research using the relationship of various semantic peers to improve the semantic analysis and resource searching. Also, we plan to implement TSPN over an actual P2P application to further evaluate and improve its performance.

References

[1] Delveroudis Y, Lekeas P V. Managing semantic loss during query reformulation in peer data management systems[J]. *Computing Research Repository*, 2010, **19**(1): 53-57.

[2] Lai G, Yang S, Yuan D. Structuring peer-to-peer networks using temporal and semantic locality[C]// *Second International Workshop on Database Technology and Applications*. Washington D C: IEEE Press, 2010.

[3] Rostami H, Habibi J, Livani E. Semantic partitioning of peer-to-peer search space[J]. *Computer Communications*, 2009, **32**(4): 619-633.

[4] Jin H, Chen H. Semrex: Efficient search in a semantic overlay for literature retrieval[J]. *Future Generation Comp Syst*, 2008, **24**(6): 475-488.

[5] Jung J J. Semantic optimization of query transformation in semantic peer-to-peer networks[C]// 2*nd International Conference on Computational Collective Intelligence-Technologies and Applications*. Washington D C: IEEE Press, 2010.

[6] Tang C, Xu Z, Mahalingam M. Psearch: information retrieval in structured overlays[J]. *Computer Communication Review*, 2003, **33**(1): 89-94.

[7] Mawlood-Yunis A R, Weiss M, Santoro N. From P2P to reliable semantic P2P systems[J]. *Peer-to-Peer Networking and Applications*, 2010, **3**(4): 363-381.

[8] Stoica I, Morris R, Karger D R, *et al*. Chord: A scalable peer-to-peer lookup service for internet applications[C]// *ACM SIGCOMM* 2001 *Conference on Applications*, *Technologies*, *Architectures*, *and Protocols for Computer Communication*. San Diego:ACM Press, 2001.

[9] Breslau L, Cao P, Fan L, *et al*. Web caching and zipf-like distributions: Evidence and implications[C]// *Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies*. Washington D C: IEEE Press, 1999.

[10] Backx P, Wauters T, Dhoedt B, *et al*. A comparison of peer-to-peer architectures[C]// *Eurescom* 2002 *Powerful Networks for Profitable Services*. Heidelberg: Springer-Verlag, 2002.

□