Journal of Southeast University (English Edition) Vol. 23, No. 3, pp. 322 – 326 Sept. 2007 ISSN 1003—7985

Semantic overlay network for searching taxonomy-based data sources

Qiao Baiyou Wang Guoren Xie Kexin

(College of Information Science and Engineering, Northeastern University, Shenyang 110004, China)

Abstract: Distributed data sources which employ taxonomy hierarchy to describe the contents of their objects are considered, and a super-peer-based semantic overlay network (SSON) is proposed for sharing and searching their data objects. In SSON, peers are dynamically clustered into many semantic clusters based on the semantics of their data objects and organized in the semantic clusters into a semantic overlay network. Each semantic cluster consists of a super-peer and more peers, and is only responsible for answering queries in its semantic sub-space. A query is first routed to the appropriate semantic clusters by an efficient searching algorithm, and then it is forwarded to the specific peers that hold the relevant data objects. Experimental results indicate that SSON has good scalability and achieves a competitive trade-off between search efficiency and costs.

Key words: peer to peer (P2P); taxonomy hierarchy; semantic searching

At present there are many distributed data sources which employ taxonomy hierarchies to describe the contents of their objects, such as PC files, web directories, and so on. Among these large-scale distributed data sources the question is how to efficiently share and exchange data information. An effective solution is to construct P2P semantic overlay networks (SONs)[1–3]. Ref. [1] first gave the idea of SONs, and proposed to classify peers and queries by taxonomy hierarchy; peers classified into a concept in the taxonomy hierarchy form an SON. But the authors gave no further details about efficient searching and storing taxonomy hierarchy. Ref. [4] proposed a DHT-based P2P network, in which the concepts in a taxonomy hierarchy are hashed into different super-peers, and peers are connected to their super-peers according to the semantics of their data objects. Super-peers are connected into a chord ring. Using consistent hash will result in loose semantics relationship of concepts in a super-peer, and make the communication among super-peers increase, and thus affect network performance. Other P2P networks, such as Edutella[5], Hypercup[6], etc. do not consider load-balancing among super-peers. We consider the characteristics of a data semantic space to consist of a taxonomy hierarchy, and present a super-peer-based semantic overlay network (SSON) which takes advantage of semantic information of the taxonomy hierarchy and the benefits of super-peer infrastructure[7]. In SSON, peers containing similar content are dynamically clustered into a semantic cluster, and different semantic clusters are organized into a semantic overlay structure, each semantic cluster only responsible for answering queries in its semantic sub-space. A query is first routed to the appropriate semantic cluster by an efficient searching algorithm, and then forwarded to the peers that hold the relevant data.

Compared to traditional super-peer networks, SSON employs a source locating strategy based on a taxonomy hierarchy, and sends queries only to the semantic clusters that satisfy the constraints of the query context. Thus, peers involved and messages to be sent are reduced and the performance of the network is greatly enhanced.

1 System Model

SSON is a hierarchy structure consisting of two layers. The bottom layer is composed of peers and the top layer is composed of super-peers. Super-peers are connected in an overlay structure according to their data semantics. Peers having contents similar to those of their super-peers together form a semantic cluster. Data are stored on peers with data indices and routing information is stored on super-peers. The intra semantic cluster data communication takes place via direct peer-to-peer links; inter semantic cluster communication takes place via links between super-peers. SSON combines the advantages of both unstructured and centralized systems. Fig. 1 is the system model of SSON.

SSON uses a common taxonomy hierarchy to classify and organize its data objects, the taxonomy hi-

Received 2007-05-18.

Foundation items: The National Natural Science Foundation of China (No. 60573089), the Natural Science Foundation of Liaoning Province (No. 20052031), the National High Technology Research and Development Program of China (863 Program) (No. 2006AA09Z139).

Biographies: Qiao Baiyou (1970—), male, graduate; Wang Guoren (corresponding author), male, doctor, professor, wanggr@mail.neu.edu.cn.

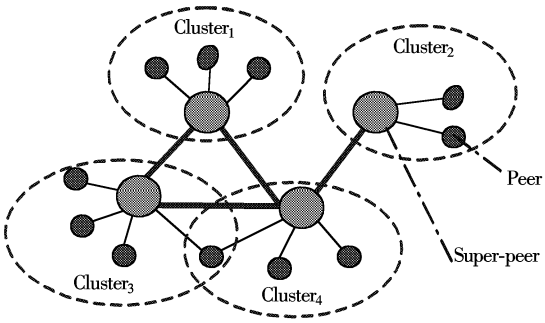

Fig. 1 The system model of SSON

erarchy represents the data semantic space, and a sub-taxonomy hierarchy stands for a semantic sub-space. A taxonomy hierarchy can be represented by a tree, called a taxonomy tree. Each semantic cluster takes charge of a semantic sub-space, comprised of one or multi sub-trees of the taxonomy tree. The index items in a semantic cluster are organized and stored in the form of one or multiple sub-trees. Each query is also classified into a node of the taxonomy tree; the node represents the semantics of the query. The query is routed according to its semantics. Each node of the taxonomy tree represents a concept and has a unique ID.

2 Constructing Approach of SSON

2.1 Constructing strategy

We apply a strategy of partitioning the data semantic space to construct SSON, in which a preset maximal cluster load size M is used to determine the clusters ranges. The chief idea is: Suppose that there is one semantic cluster in the network initially, whose semantic space is the whole taxonomy tree; the load size of the cluster increases with peers joining. If the load size of the cluster exceeds M, the cluster will automatically partition its semantic sub-space according to the data semantics and the load, and a new semantic cluster is generated from the original one. The new semantic cluster selects some appropriate semantic clusters as its neighbors and establishes routing links, and is responsible for answering queries in its semantic sub-space. In this way, SSON is formed. The advantage of the strategy is that the load size of a cluster is less than the maximal load size M, and, hence, search efficiency of the system is guaranteed. Therefore, the approach is adaptive to differences in density of peers in the semantic space, and ensures stability and adaptability of the system.

2.2 Clustering algorithm

In SSON, any super-peer, as an index server, takes charge of vast work such as maintaining indices, answering and routing queries, etc., and thus the capability of super-peers determines the network performance. Therefore, we regard the load of a super-peer as the load of the cluster, which is the important basis of our clustering algorithm. Since maintenance and search take a super-peer a lot of time, the load of a super-peer can be measured by the number of index items. Thus we can express a semantic cluster with a set of weighted taxonomy sub-trees, the weight of a node representing the number of index items, and the topological relationship between nodes representing the semantic relationship between the corresponding index items. Now the clustering problem can be regarded as a partitioning of a set of weighted taxonomy sub-trees. When clustering, we should consider not only the semantics between index items but also the load-balancing among super-peers, and, in fact, it is a tradeoff between load-balancing among clusters and semantics within a cluster. Focusing on the problem, we propose a self-organized semantic clustering algorithm, i.e. LBFCA[8], which achieves a good trade-off between data semantics within a cluster and load-balancing among clusters.

2.3 Semantic cluster encoding strategy

In SSON, each cluster has a unique ID, representing the location of the cluster in the whole semantic space and being the basis of establishing neighbor links. This paper offers a cluster encoding algorithm, which automatically generates an ID for a new cluster. The system uses a binary number of m bits as a cluster ID, and each cluster maintains a variable Par_times to record the partitioning time. The first cluster ID is predefined as 0 and Par_times as 0. When the cluster is partitioned with peers joining, the newly generated cluster obtains a new ID that equals the ID of the original cluster plus $2^{m-\text{Par\_times}-1}$, and Par_times is increased by one. When Par_times is more than m, partitioning can no longer be done, and therefore m should be large enough. The cluster encoding algorithm is shown as follows:

Algorithm 1 Algorithm for cluster encoding

Input: ID_{old}: ID of the original cluster;
Par_times: partitioning times;
m: binary length of a cluster ID.
Output: ID_{new}: ID of the new cluster.
if Par_times < m then
$\text{ID}_{\text{new}} = \text{ID}_{\text{old}} + 2^{m-\text{Par\_times}-1}$
Par_times = Par_times + 1
End if

2.4 Construction of the semantic clusters overlay

In SSON, each cluster maintains a cluster information table that stores relevant status and routing information. There are seven fields: Cluster_ID is the cluster ID; Par_times is the partitioning time; Cluster_size is the load size of the cluster, measured by the amount

of index entries or peers; Cluster_range is current semantic sub-space of the cluster, represented by a set of IDs of root nodes of taxonomy sub-trees; Network_range is initially semantic sub-space of the cluster when the cluster is generated, represented by a set of IDs of root nodes of taxonomy sub-trees, representing semantic sub-spaces of the cluster and those partitioned apart from it; Ancestor_link is a link to its ancestor neighbor cluster, comprised of Network_range and Cluster_ID of its ancestor neighbor cluster; Ordinary_links is a set of neighbor links that point to non-ancestor neighbor clusters, each neighbor link consisting of Network_range and Cluster_ID of the neighbor cluster. Each cluster maintains some links to its neighbor clusters. In SSON, the neighbor clusters are established in the way that two clusters are neighbors if their IDs are one bit different. According to the cluster encoding method, a newly partitioned cluster and the original one must be neighbors. The neighbor clusters are established as below. When the cluster A generates a new cluster B, their cluster information tables are updated respectively. If there is an ID in A's network_range that has an ancestor-descendant relationship with an ID of B's Network_range, A is B's ancestor neighbor and B is A's ordinary neighbor. Otherwise, they are ordinary neighbors of each other, and meanwhile A's ancestor neighbor is B's ancestor neighbor. At the same time, B broadcasts its ID to find clusters whose IDs are one bit different from B's, and if such clusters are found within certain steps, their information is added to the Ordinary_links of all the others and they become ordinary neighbors. Obviously, the method of establishing neighbor clusters makes each cluster have several ordinary neighbor clusters and have an ancestor neighbor cluster, but SC_0 only has ordinary neighbor clusters. This method can reduce the search hops and communication cost, and therefore it improves query performance of the system.

For example, as shown in Fig. 2, initially, there is one cluster SC_0, consisting of a taxonomy tree. With load increase, SC_8 is firstly partitioned from the cluster SC_0, then SC_0 and SC_8 continue being partitioned with load increase; SC_{12} is partitioned from SC_8, SC_4 and SC_6. After several times of clustering, the original cluster is clustered into five clusters. Accordingly, the taxonomy tree is partitioned into several taxonomy sub-trees, and each cluster consists of one or multi taxonomy sub-trees. According to the mechanism of neighbor cluster establishment, the neighbor links among the five clusters and their cluster information are shown in Fig. 3.

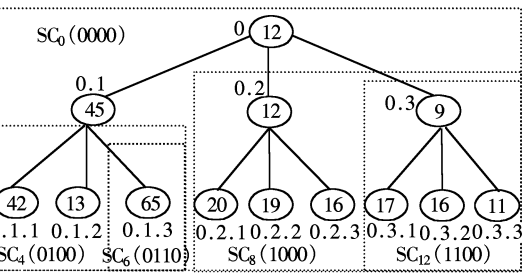

Fig. 2 A clustering example

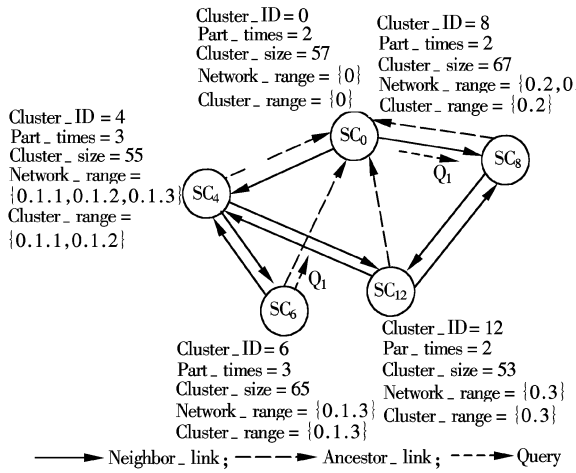

Fig. 3 Neighbor relationship among five clusters

3 Searching Algorithm

Since the semantics of data object and query are described with taxonomy hierarchy, each query is associated with a concept of the taxonomy hierarchy, which corresponds to a node on the taxonomy tree, the node expressing the semantics of the query. We name the ID of the node as search ID (SID). According to the structural characteristics of the taxonomy tree, when a node is the searching target of a query, its ascendants and descendants are searching targets as well, and hence the query is first sent to the semantic clusters that contain the target nodes. In this paper, the ascendant-descendant relationship is judged by the node IDs of the taxonomy tree. The searching process is as follows.

When a peer in the cluster C sends a query Q to its super-peer, the super-peer obtains the SID from Q's semantic information. If the SID has an ascendant-descendant relationship with some ID(s) in C's Cluster_range, Q is first sent to C's ancestor neighbor cluster A by Ancestor_link, and then C executes query Q, finds all the peers in its index that can answer Q, sending Q to these peers to execute and return results to the peer that initially sends Q. At the same time, it is checked whether in each Network_range in C's Ordinary_links, there is some ID of ascendant-descendant relationship with the SID. If so, Q is first sent to these neighbor clusters. If in C's Cluster_range and any Net-

work_range in the Ordinary_links, there is no node whose ID has an ascendant-descendant relationship with SID, Q is forwarded to the ancestor neighbor cluster A by Ancestor_link. The detail searching algorithm is shown as follows:

Algorithm 2 Searching algorithm

```
Input: SID: the query's search ID;
If received search(SID) before then
   Drop search(SID);
   Return;
end if
Search_token = 0;
If exist a ascendant-descendant relationship between SID and a NID
in cluster_range then
   Forward search(SID) to ancestor neighbor cluster Ancestor_link.
   Cluster_ID
   local_index_search(SID);
   Search_token = 1;
End if
For any neighbor link NL in Ordinary_links do
If exist a ascendant-descendant relationship between SID and a NID
in NL. Network_range then
   Forward search(SID) to neighbor cluster NL. Network_range.
   Cluster_ID
   Search_token = 1;
   End if
End for
If Search_token! = 1 then
   Forward search(SID) to ancestor neighbor cluster
   Ancestor_link. Cluster_id;
End if
```

As shown in Fig. 3, a peer in the cluster SC_6 sends a query Q_1 whose SID is 0. 2. 2. According to algorithm 2, the searching path is shown in the form of arrows. Since no ID in SC_6's Cluster_range and none of Network_range in its Ordinary_links have ascendant-descendant relationship with the SID, Q_1 is routed to its ancestor neighbor cluster SC_0. According to its Cluster_range, SC_0 is recognized as one of the targets of Q_1. The ID 0. 2 in Network_range in SC_0's Ordinary_links has an ascendant-descendant relationship with the SID, and then Q_1 is forward to the neighbor SC_8, and thus SC_8 is another target of Q_1.

4 Performance Evaluation

We evaluate the performance of SSON using simulating experiments. Synthesized data are employed to test the performance of the algorithm. There are many taxonomy trees of 2 to 10 layers composed of 200 to 100 K concepts (nodes) of 2 to 30 fan-outs. For calculation convenience, it is assumed that each peer relates to only one index item, and thus the load size of a cluster can be viewed as the number of peers. For universality, the distribution of peers on the taxonomy tree follows the Zipf-like ($\alpha = 0.8$) distribution. At the beginning there is one cluster and one super-peer in the system. With peers continuously joining, the cluster is automatically partitioned when the maximal cluster load size M is exceeded. We made a comparison between SSON and DHT-based network[4], the results are as follows.

In this paper, the average query hops (AQH) is used to evaluate query efficiency and average message number per query (AMNPQ) to evaluate query cost. We construct a four-layered taxonomy tree composed of 3 204 concepts. When the maximal cluster load size M is 100 and 50 queries are sent randomly from each peer, the AQH with network size varying are shown in Fig. 4. It can be seen that the AQH increases slowly with network size increasing, and the AQH of the SSON is lower than that of DHT-based network. Since SSON clusters peers based on their data semantics and routes semantically, and thus reduces the communications among cluster. The AQH consequently decreases. The DHT-based network cannot support semantic clustering and thus data semantics in a cluster are loose, so the AQH is higher. Fig. 5 shows how the AMNPQ changes in the same condition as that in Fig. 4. It can be seen that the AMNPQ slowly increases with peers increasing almost consistently with the changing in the AQH. From both aspects, it can be seen that SSON is better.

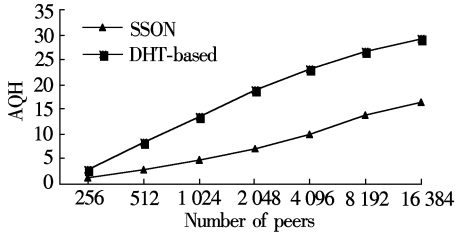

Fig. 4 Comparison of AQH with the network size varying based on SSON and DHT-based network

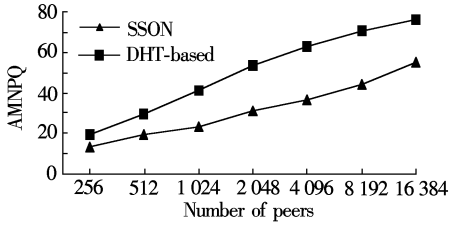

Fig. 5 Comparison of AMNPQ with the network size varying based on SSON and DHT-based network

With the same condition as that in Fig. 4, and when the maximal cluster load size M equals 100, 200 and 400; and with the network size varying the results of AQH and AMNPQ based on SSON and the DHT-based network are shown in Fig. 6 and Fig. 7, respec-

tively. It can be seen that when M is large, the AQH and AMNPQ decrease, this is because when M is large, the total number of clusters in the system decreases, thereby reducing queries across clusters and messages to be forwarded. But the load within a cluster is also increased, so M should be suitable.

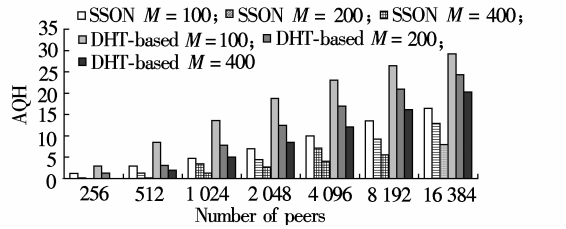

Fig. 6 Results of AQH with the network size varying based on SSON and DHT-based network when M varies

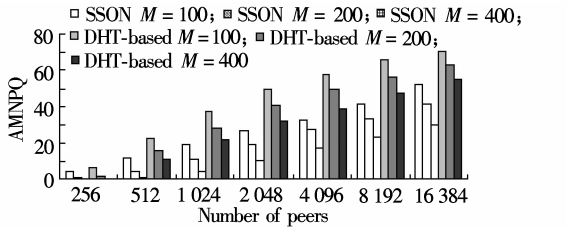

Fig. 7 Results of AMNPQ with the network size varying based on SSON and DHT-based network when M varies

5 Conclusion

In this paper, we focus on how to construct a semantic overlay network using semantic information contained by a domain taxonomy hierarchy, and propose SSON, a super-peer based semantic overlay network. We also give relevant algorithms and make a comparison with the DHT-based network. Experiments show that SSON has low search latencies and overheads. Future work focuses on the dynamical load-balancing algorithms among semantic clusters and maintenance approaches in the case of super-peer departure and failure based on improving clustering and the searching algorithm.

References

[1] Crespo A, Garcia-Molina H. Semantic overlay networks for P2P systems[R]. Stanford University, 2003.

[2] Löser A, Tempich C. On ranking peers in semantic overlay networks[C]//*The* 3*rd Conference on Professional Knowledge Management* (*PAIKM* 2005). Kaiserslautern, Germany, 2005: 209 - 216.

[3] Löser A, Naumann F, Siberski W, et al. Semantic overlay clusters within super-peer networks[C]//*Proceedings of the International Workshop on Databases, Information Systems and Peer-to-Peer Computing in Conjunction with the VLDB*. Berlin, Germany, 2003: 33 - 47.

[4] Löser A. Towards taxonomy based routing in P2P networks [C]//*Workshop on Semantics in Peer-to-Peer and Grid Computing on the* 13*th WWW Conference*. New York, 2004: 407 - 412.

[5] Nejdl W, Wolf B, Qu C, et al. EDUTELLA: a P2P networking infrastructure based on RDF[C]//*Proceedings of the* 11*th International WWW Conference*. Hawaii, USA, 2002: 604 - 615.

[6] Nejdl W, Wolpers M, Siberski W, et al. Super-peer-based routing and clustering strategies for RDF-based P2P networks[C]//*Proceedings of the* 12*th International WWW Conference*. Budapest, Hungary, 2003: 536 - 543.

[7] Yang B, Garcia-Molina H. Designing a super-peer network [C]//*Proc of the* 19*th International Conference on Data Engineering*. Bangalore, India, 2003: 49 - 74.

[8] Qiao Baiyou, Wang Guoren, Xie Kexin. A self-organized semantic clustering approach for super-peer networks[C]//*Web Information Systems-WISE*2006. Wuhan, China, 2006: 448 - 453.

一种支持分类数据源查找的语义覆盖网络

乔百友　王国仁　谢可心

(东北大学信息科学与工程学院,沈阳 110004)

摘要:针对使用分类层次来描述数据语义的分布式数据源,提出了一种支持数据共享的基于 super-peer 的语义覆盖网络 SSON. SSON 能够根据数据的语义,动态地将 peer 划分成多个语义簇,语义簇之间组织成语义覆盖网络. 每个语义簇由一个 super-peer 和一组 peer 组成,仅负责回答其语义子空间上的查询. 查询首先根据其语义被路由到适合的语义簇中,然后被转发给包含结果的 peer. 同时给出了相关的算法,并进行了实验研究,实验结果表明,SSON 具有良好的可扩展性,并在查找性能和代价之间取得了一个良好的折中.

关键词:P2P;分类层次;语义查找

中图分类号:TP393