AN INDEXING METHOD FOR SUPPORTING SPATIAL QUERIES IN STRUCTURED PEER-TO-PEER SYSTEMS

Lingkui Meng [a], Wenjun Xie [a, *], Dan Liu [a, b]

[a] School of Remote Sensing and Information Engineering, Wuhan University, Wuhan 430079, China
[b] Department of Information Technology, Huazhong Normal University, Wuhan 430079, China
(lkmeng, xiewenjun)@whu.edu.cn, rock_fall@tom.com

Commission VI, WG VI/4

KEY WORDS: Peer-to-Peer, Spatial Queries, Overlap Minimization, Spatial Index, Tree Structure, Distributed Optimization

ABSTRACT:

To provide the efficient supporting spatial data queries in peer-to-peer systems has recently received much attention. Most proposed methods tried to use hop count to represent the transmission delay, and the total message count to estimate the cost of query processing. For the ignorance of the differences between DHT lookups and spatial queries, and distinction between physical networks and overlay networks, the efficiency and cost of their query processing can't be indicated properly. In addition, their experimental results are achieved by using point data sets, while the fact that the overlap of spatial objects usually exists in real applications is not considered, and it may cause multi path query processing and then results in plenty of peers visiting and routing messages. In this paper, we propose an indexing method which efficiently supports spatial queries in structured peer-to-peer systems. It adopts an overlap minimization algorithm which takes the query rate of data into account to reasonably reduce the holistic cost of queries. We also introduce a dynamically adaptive distributed optimization scheme that dynamically adapting to the time-varying overlay architecture and data usage concerns. Theoretical analysis and simulation results both indicate that our method is efficient and effective.

1. INTRODUCTION

With the rapid growth and increased importance of distributed spatial data, there is an increasing need for massive spatial data sharing in large-scale distributed systems. While the P2P (Peer-to-Peer) systems have become a powerful means for data sharing in Internet community. For their potential uses in spatial data sharing, to provide the efficient supporting spatial data queries in peer-to-peer environments has recently received much attention.

There are two types of P2P overlay networks: unstructured ones in which requests are broadcasted or routed through flooding or random walks, such as Gnutella (Ripeanu, 2001), KaZaA (Leibowitz, 2003), eDonkey (Tutschku, 2004), Freenet (Clarke, 2000), and structured ones based on DHTs (Distributed Hash Tables) in which requests are routed using routing tables, such as Chord (Stoica, 2003), CAN (Ratnasamy, 2001), Tapestry (Zhao, 2004), Pastry (Rowstron, 2001). Since they use flooding or random walks based methods for processing queries request, which results in a large number of messages and a traffic overhead, so lead to the poor efficiency of queries and the bad scalability for the systems, and these restricted their rapid development. On the other hand, it is very easy to process a query by using the assigned key in structured P2P networks, and their scalability is very good. So the structured P2P systems are more appropriate for handling data sharing. But for the reason of using DHTs, which destroy the semantics of the data objects, only exact key match queries can be supported efficiently, and it isn't an easy task to support spatial queries in structured P2P systems.

The traditional indexing methods for spatial databases can be briefly classified into two approaches: spatial sorting-based, such as Hilbert space filling curve (Bially, 1969), Z-ordering curve (Orenstein, 1986), and spatial contains relationship-based, such as R-tree (Guttman, 1984), R+-tree (Sellis, 1987), R*-tree (Beckmann, 1990). Similarly, the methods that support spatial data queries in structured P2P systems also have two kinds of implements: the one that maps multidimensional spatial data into one-dimensional by using order-preserving hash function, and the other one that distribute tree data structure in P2P environment. The main problem of former approach is the spatial relationships between spatial objects often may be destroyed, so leads to the inefficiency of the queries. This is because there are no any functions can always preserve the spatial properties. As for the latter method, a critical performance issue is the tree structure has to be queried in a top-down manner from the root node, so the communication bottlenecks are more likely to happen on the peers that take charge of the tree nodes at higher level, especially for root node, and it is also a single point of failure. For their good efficiency in centralized environment and the bad performance of the former approach, using hierarchical tree structure is a better choice. VBI-Tree (Jagadish, 2006) solved the above problem in latter approach by introduce a new routing table design using sideway index links, and DPTree (Li, 2006) handled it by propose tree branch oriented distribution.

* Corresponding author.

Although some proposed methods have achieved good results. There still exist some issues that are not considered carefully. 1) The differences between DHT lookups and spatial queries. In a DHT lookup, the query request is forwarded along a single path, while it may be routed over multiple paths for spatial query. The multiple peers visiting sometimes can't be avoided. However, we can decrease the number of multiple peers visiting. 2) The distinction between physical networks and overlay networks. So the hop count can not really reflect the transmission delay.

To address the above two issues, we propose a suite of efficient solutions, which can be used to support any kind of hierarchical tree architecture overlay. Our paper makes the following two major contributions. 1) An innovative definition of overlap is first introduced. Here we take the following properties of the systems into consideration: the non-uniform and time-varying properties of spatial data distribution and their popularity, and peer interests also are different and time-varying. Then we present an overlap minimization algorithm to minimize the number of peers need to visit for process a query. 2) We propose an efficient distributed optimization algorithm to guarantee each peer has neighbors that are physically close to it in the underlying network, and it can continuously and efficiently optimize the overlay structure under dynamic network conditions.

The rest of the paper is organized as follows. Section 2 surveys previous work, focusing on spatial data queries in structured P2P systems. In section 3 we propose the definition of overlap and the overlap minimization algorithm. The distributed optimization algorithm is explained in section 4. The experimental results of the above design, using many metrics, such as routing hop count, number of messages, distribution percentage of delay time of peers, are presented in section 5. Finally, we discuss the conclusions and future work in section 6.

2. RELATED WORK

There has been a substantial amount of research on spatial data queries in P2P systems. As mentioned, most current proposed methods generally can be briefly classified into two categories: one is the reduction of multidimensional spatial data to one dimension, then the current DHTs can be directly used to support spatial queries, and the other is distribution of spatial contains relationship-based indexing structure.

The former category includes SCRAP (Ganesan, 2004), MAAN (Cai, 2004), PRoBe (Sahin, 2005). SCRAP uses a two-step solution to partition the data space. In the first step, it map multi-dimensional spatial data down to one-dimensional by using a space-filling curve, then the one-dimensional data could be range partitioned across the dynamic available peers. MAAN supports multi-attribute range queries through multiple single-attribute resolution by using locality preserved hashing to map a range of data space to Chord, and the efficiency may be very poor. PRoBe uses a multi-dimensional logical space and maps data items onto the space based on their attribute values, and the space is divided into hyper-rectangles, with each maintained by a peer within the system.

In the latter category, VBI-Tree is an abstract data structure build based on a virtual binary balanced tree structure. It was inspired by BATON (Jagadish, 2005) structure where each peer corresponds an internal node and a leaf node. DPTree uses a tree branch oriented distribution method to distribute the tree structure among peers in a way preserving the good properties of balanced tree structures yet avoiding single points of failure and performance bottlenecks.

It is known that tree structures are very difficult to distribute in P2P systems, because searching the tree by following paths induces an uneven load on tree nodes at higher level. The above two methods solve the problem by introduce some novel designs. However, the efforts only aim to support zero dimensional data queries in multi-dimensional data space, and none of them consider the situations of multi-dimensional data in multidimensional data space. More specifically, the current methods can not efficiently handle the queries about line data or polygon data, since they omitted the overlap between adjacent tree nodes, which results in the multiple peers visiting in distributed systems. In addition, we should also try to keep each peer has neighbors that are physically close to it in the underlying network.

3. OVERLAP MINIMIZATION

For the existence of overlap in tree nodes, there will be multiple paths need to be followed even for point data queries, which results in a large number of messages and plenty of peers visiting. We therefore first give our definition of overlap in P2P systems, and then present an overlap minimization algorithm.

3.1 Definition of Overlap

In the X-tree method (Berchtold, 1996), the following two definitions of overlap are given.

If a tree node contains *n* hyper rectangles $\{R_1, \ldots, R_n\}$, the overlap can formally be defined as

$$Overlap = \frac{\left\| \bigcup_{i,j \in \{i \ldots n\}, i \neq j} (R_i \cap R_j) \right\|}{\left\| \bigcup_{i \in \{1 \ldots n\}} R_i \right\|} \quad (1)$$

where $\|A\|$ denotes the volume covered by A.

For the distribution of spatial data is nonuniform, the modified definition took this into account. That is

$$WeightedOverlap = \frac{\left| \{ p \mid p \in \bigcup_{i,j \in \{i \ldots n\}, i \neq j} (R_i \cap R_j) \} \right|}{\left| \{ p \mid p \in \bigcup_{i \in \{1 \ldots n\}} R_i \} \right|} \quad (2)$$

where $|A|$ denotes the number of data elements contained in A.

Obviously, the popularity between data elements is different. A more accurate definition of overlap needs to take the query rate of data elements into account. So we propose the following new definition.

$$NewOverlap = \frac{q_p \times \left| \{ p \mid p \in \bigcup_{i,j \in \{i \ldots n\}, i \neq j} (R_i \cap R_j) \} \right|}{q_{all} \times \left| \{ p \mid p \in \bigcup_{i \in \{1 \ldots n\}} R_i \} \right|} \quad (3)$$

where p is a data element
q_p denotes the query rate of p, which is the number of received queries about p during the history period.
q_{all} denotes the query rate of all data elements, which is the number of received queries about all data elements stored in the local peer during the history period.

As data element popularity and peer interest is time-varying, we use exponential moving average technique to calculate the value of query rate, which give more weight to the observations in most recent periods without discarding other values, rather than directly using the observation values during the entire history period. So the new formula for calculating query rate is as follows.

$$\overline{q}_{curr} = w \times \overline{q}_{prev} + (1 - w) \times q_{curr} \quad (4)$$

where $\overline{q}_{curr}$ is current valid value for query rate
$\overline{q}_{prev}$ is the previous valid one
q_{curr} is the current observed one
$w \in [0,1]$ is a constant value which represents a weight factor value for new observation.

3.2 Overlap Minimization Algorithm

If an overlap will appear on one peer, we should choose to allow its existence or adjust the position of some influenced peers to minimize the overlap. The latter action will be taken only when the benefits brought about by the existence of overlap is less than its cost. If the overlap occurred in a peer, and it will affect n data elements $\{d_1, d_2, \ldots, d_n\}$, with the fractions of area of the overlap to them are $\{f_1, f2, \ldots, f_n\}$, and the average cost to process query about these data elements are $\{c_1, c_2, \ldots, c_n\}$. We also assume it will cost some extra system resources SR to allow the existence of the overlap. If the benefits of the overlap less than its cost, then we keep the overlap, or adjust some peers. That is

$$\sum_{i=1}^{n} f_i \times q_i \times c_i < SR \quad (5)$$

where q_i is the query rate of d_i.

4. DISTRIBUTED OPTIMIZATION ALGORITHM

We define the delay time of a peer p as ***D(p)***, which is the sum of latencies to all its neighbours. Its definition can formally be defined as

$$D(p) = \sum_{n \in neighbors(p)}^{|neighbors(p)|} L(p,n) \quad (6)$$

where p is a peer
neighbours (p) is the set of *p*'s neighbours
n is one of its neighbours
L*(p, n)* is the latency of peer p to n.

It is easy for us to know that the optimum situation is to keep the minimum value of total delay time $D_{sum}(p)$ of all peers S involved in the system.

$$D_{sum}(p) = \sum_{p \in S}^{|S|} D(p) \quad (7)$$

where S is the set of all peers involved in the system

Since in the P2P systems it is impractical to calculate the above value, we propose a distributed optimization algorithm that is an iterative algorithm that each peer executes periodically and uses integer linear optimization to minimize the sum of the delay time of a peer and one of its neighbours.

The algorithm is executed periodically on two adjacent peers in the system, which we call them *seeds*. Firstly, they mutually exchange the routing table of their neighbours, and then each seed peer measures the latencies to the neighbours of the other seed peer. Finally, we can determine whether or not to swap their neighbours based on the measured values. The function that we want to minimize during the iteration of the algorithm is as follows.

$$\sum_{p \in Seeds}^{|Seeds|} D(p) = \sum_{p \in Seeds}^{|Seeds|} \sum_{n \in neighbours(p)}^{|neighbours(p)|} L(p,n) \quad (8)$$

where *Seeds* are two adjacent peers in the system

It can be proved that global convergence can be achieved but proofs are omitted due to space limitations.

5. EXPERIMENTAL RESULTS

For the evaluation of the above two algorithms we did compare the experimental results between our optimized method and the original one in a tree structure overlay. Since the structure here didn't consider the properties of the physical network, we use an Internet node latency measurement results (Wong, 2005) from Meridian project in Cornell University. As there is only point data simulator component in the design, we also use a spatial data generator (University of Piraeus, 2006) to create polygon data, which are Zipf distribution.

We test the network with different number of nodes N from 100 to 2500. For each test, 50 point queries are executed, and then the average value is taken. In Figure 1 we present the results of number of messages to locate data to process point queries for point data and polygon data, and the number can be used as a metric of scalability of the system. As we observe that our algorithm could reduce the query processing cost for polygon data, but nearly the same as original algorithm for point data.

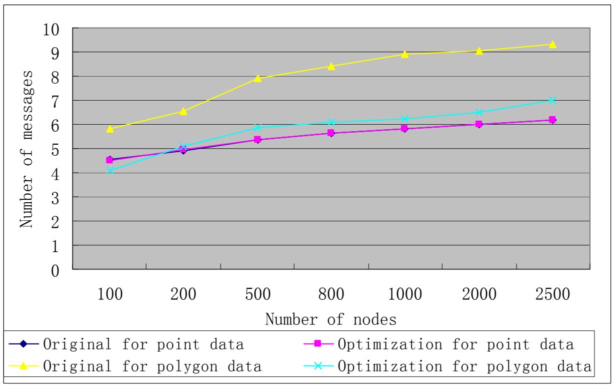

Figure 1. Number of message comparison

In Figure 2 we demonstrate the number of routing hops comparison, which can reflect the efficiency of a system. Similar as the above results, the figure indicated that our algorithm could significantly improve the efficiency for polygon data, but not for point data

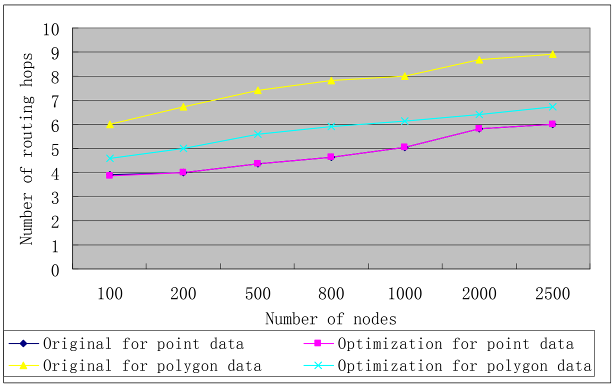

Figure 2. Number of routing hops comparison

Figure 3 shows the results of distributed optimization algorithm for keep adjacent peers in physical network as neighbors in overlay network, which prove that the method has good locality properties for tree-based overlay networks.

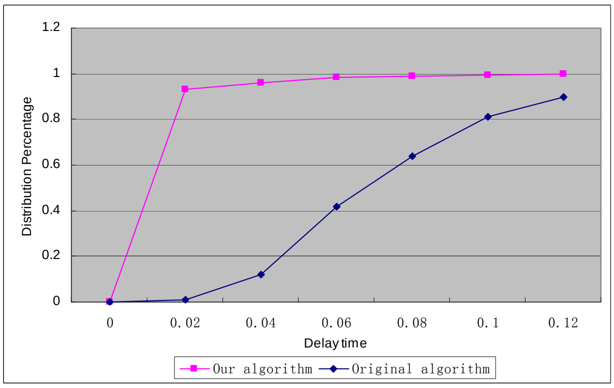

Figure 3. Distribution percentage of delay time of peers

6. CONCLUSION AND FUTURE WORK

In this paper, we proposed our indexing method for supporting spatial queries in P2P systems, which allows us to reduce the cost of routing messages and improve the efficiency of routing hops. Additionally, by recognizing that hop count can not reflect the actual time required for processing queries, we also focus on the reduction of the delay time of each hop besides the hop count, and hence decrease the total time. The method is based on two newly proposed algorithms: overlap minimization algorithm and distributed optimization algorithm.

For the future, we will augment our method to include other efficient query algorithms, such as range query and kNN query. Upon completion of this work, we also plan to run comprehensive performance evaluation on our method.

ACKNOWLEDGEMENTS

The research is supported by Natural Science Foundation of China (40771177).

REFERENCES

Ripeanu, M. 2001. Peer-to-Peer Architecture Case Study: Gnutella Network. *1st International Conference on Peer-to-Peer Computing (P2P 2001)*, Linköpings universitet, Sweden, pp. 99-100.

Leibowitz, N., Ripeanu, M., Wierzbicki, A. 2003. Deconstructing the Kazaa network. *3rd IEEE Workshop on Internet Applications (WIAPP 2003)*, San Jose, California, USA, pp. 112-120.

Tutschku, K. 2004. A Measurement-based Traffic Profile of the eDonkey Filesharing Service. *5th Passive and Active Network Measurement Workshop (PAM 2004)*, Antibes Juan-les-Pins, France, pp. 12-21.

Clarke, I., Sandberg, O., Wiley B., et al. 2000. Freenet: A Distributed Anonymous Information Storage and Retrieval System. *Workshop on Design Issues in Anonymity and Unobservability 2000*, Berkeley, California, USA, pp. 46-66

Stoica, I., Morris, R., et al. 2003. Chord: a scalable peer-to-peer lookup protocol for Internet applications. *IEEE/ACM Transactions on Networking*, 11(1), pp. 17–32.

Ratnasamy, S., Francis, P., et al. 2001. A Scalable Content-Addressable Network, *ACM SIGCOMM 2001*, San Diego, California, USA, pp. 161-172.

Zhao, B. Y., Huang, L., Stribling, J., et al. 2004. Tapestry: A Resilient Global-Scale Overlay for Service Deployment, *IEEE Journal on Selected Areas in Communications*, 22(1), pp. 41-53.

Rowstron, A. I. T., and Druschel, P. 2001. Pastry: Scalable, Distributed Object Location and Routing for Large-Scale Peer-to-Peer Systems. *18th IFIP/ACM International Conference on Distributed Systems Platforms*. Heidelberg, Germany, pp. 329-350.

Bially, T. 1969. Space-Filling Curves: Their Generation and Their Application to Bandwidth Reduction. *IEEE Transactions on Information Theory*, 15(6), pp. 658-664

Orenstein, J. 1986. Spatial Query Processing in an Object-Oriented Database System, ACM SIGMOD International Conference on Management of Data, Washington, D. C., USA, pp. 326-336,

Guttman, A. 1984. R-trees: A dynamic index structure for spatial searching. *1984 ACM SIGMOD Internation Conference on Management of Data*, Boston, Massachusetts, USA, pp. 47-57.

Sellis, T., Roussopoulos, N., Faloutsos, C. 1987. The R+-Tree: A Dynamic Index for Multidimensional Objects. *13th International Conference on Very Large Data Bases (VLDB 1987)*, Brighton, England, pp. 507-518

Beckmann, N., Kriegel, H. P., et al. 1990. The R*-tree: An efficient and robust access method for points and rectangles. *ACM SIGMOD 1990*, Atlantic City, New Jersey, USA, pp. 322-331.

Jagadish, H. V., B. C. Ooi, et al. 2006. VBI-Tree: A Peer-to-Peer Framework for Supporting Multi-Dimensional Indexing Schemes. *22nd International Conference on Data Engineering (ICDE 2006)*. Atlanta, Georgia, USA, pp. 34.

Li, M., W.-c. Lee, et al. 2006. DPTree: A Balanced Tree Based Indexing Framework for Peer-to-Peer Systems. *14th IEEE International Conference on Network Protocols (ICNP 2006)*. Santa Barbara, California, USA, pp. 12-21.

Ganesan, P., Yang, B., et al. 2004. One Torus to Rule them All: Multidimensional Queries in P2P Systems. *7th International Workshop on the Web and Databases (WebDB 2004)*, Paris, France, pp. 19-24.

Cai, M., Frank, M., Chen, J., et al. 2004. MAAN: A Multi-Attribute Addressable Network for Grid Information Services. *Journal of Grid Computing*, 2(1), pp. 3-14.

Sahin, O. D., Antony, S., Agrawal, D., et al. 2005. PRoBe: Multi-Dimensional Range Queries in P2P Networks. *6th International Conference on Web Information Systems Engineering (WISE 2005)*, New York City, New York, USA, pp. 332-346.

Jagadish, H. V., Ooi, B. C., Vu, Q. H. 2005. BATON: A Balanced Tree Structure for Peer-to-Peer Networks. *31st International Conference on Very Large Data Bases (VLDB 2005)*. Trondheim, Norway, pp. 661-672.

Berchtold, S., Keim, D. A., Kriegel, H. 1996. The X-tree: An Index Structure for High-Dimensional Data. *22nd International Conference on Very Large Data Bases (VLDB 1996)*. Mumbai, India, pp. 28-39.

Wong, B., Slivkins, A., Sirer, E. G. 2005. Meridian: A Lightweight Network Location Service without Virtual Coordinates. *ACM SIGCOMM 2005*, Philadelphia, Pennsylvania, USA, pp. 85-96.

Meridian Project, http://www.cs.cornell.edu/People/egs/merid-ian/

University of Piraeus. 2006. http://www.rtreeportal.org/