Available online at www.sciencedirect.com
ScienceDirect
ELSEVIER
June 2010, 17(3): 66–72
www.sciencedirect.com/science/journal/10058885
The Journal of China Universities of Posts and Telecommunications
http://www.jcupt.com

HPSIN: a new hybrid P2P spatial indexing network

WU Jia-gao[1], JIANG Nan[2] (✉), ZOU Zhi-qiang[1], HU Bin[2], HUANG Lin[1], FENG Jia-li[3]

1. Institute of Computer Technology, Nanjing University of Posts & Telecommunications, Nanjing 210003, China
2. Key Laboratory of Virtual Geographic Environment, Ministry of Education, Nanjing Normal University, Nanjing 210046, China
3. School of Geographic and Oceanographic Sciences, Nanjing University, Nanjing 210093, China

Abstract

Geographic information system (GIS) is increasingly managing very large sets of data, hence a centralized data index may not always provide the most scalable solution. Recently, the peer to peer (P2P) networks have become very popular for sharing information in a totally decentralized manner. In this paper, a new hybrid P2P spatial indexing network (HPSIN) is proposed, which combines distributed quad-tree with distributed Hash table (DHT) based Chord network to maintain both query efficiency and system load balance. In addition, a simple theoretical model based on opened queueing network for HPSIN is established. Assuming each peer as M/M/1 queueing processor in the model, fundamental characteristics of the system is captured, and expression of average query delay is obtained in close form. The theoretical analysis and numerical computing results show that there exists an optimum point of tradeoff between efficiency and load balance. By setting a proper value of start index level l_s for different network scale and query rate, HPSIN will achieve the minimum overall query delay, therefore, can adapt to different P2P application environments.

Keywords GIS, P2P networks, spatial indexing, queueing network, theory

1 Introduction

With the advancement of computing and networking technologies, GIS has evolved rapidly and become representation of new generation computer applications. One of the most fundamental challenges of GIS is how to organize and index the large scale geospatial data efficiently. Traditional spatial data indexing approaches are mainly based on one of the two kinds of tree-form structures: Quad-tree [1] and R-tree [2], which can support commonly complex spatial range queries and localize objects of interest. However, with more and more concurrent users to access geographic services and huge spatial data, client/server (C/S) based GIS unavoidably suffers with bottleneck and single point of failure at central server.

Recently, research has focused on P2P networks to handle such problem. Typically, a P2P network is a distributed environment composed of autonomous peers that operate in an independent manner. Each peer plays roles as both client and server in P2P network, which eliminates the bottleneck and shares resources and services among peers fully. There are three typical network models in P2P network. The first model is unstructured P2P network, such as Gnutella [3], Freenet [4], etc., where peers are constructed into an irregular topology randomly. This model is simple and robust, but the efficiency of query is bad because of the flooding lookup message. The second model is structured P2P network, where peers are organized into some regular topologies based on so-called DHT routing algorithm. The famous examples of DHT-based P2P networks are Chord [5], CAN [6], Pastry [7], etc. DHT routing improves the scalability and load balance of P2P network, but supports complex query poorly. The third model is hierarchical P2P network (i.e. NICE [8], ZIGZAG [9], etc.), where peers with the same semantic feature make up a cluster. In the interior of the cluster, the first model or even the simplest C/S structure is adopted. Each cluster selects a header peer who participates to constitute upper layer cluster, until the top layer cluster header or the root is elected. The hierarchical model has high efficiency, but a bit of fragile due to tree-form structure.

Received date: 10-10-2009
Corresponding author: JIANG Nan, E-mail: njiang@njnu.edu.cn
DOI: 10.1016/S1005-8885(09)60468-4

The existing P2P spatial indexing networks for GIS can be mainly classified into two paradigms. The first paradigm constructs a distributed version of the centralized tree based spatial index, which forms a hierarchical P2P networks. Mondal et al propose a hierarchical spatial indexing network called P2PR-tree, where peers compose a distributed R-tree [10]. To alleviate the workload of root peer, P2PR-tree partitions the top level minimum bounding rectangle (MBR) statically and stores the MBR information into all peers. However, the static partition constrains the adaptation and efficiency of the system. Liu et al's proposal is also based on R-tree capable of processing complex multi-dimensional queries [11]. Considering the heterogeneity of peers, in a cluster, ordinary peer only maintain local R*-tree, whereas super peer maintain global inter cluster index with network-R-tree (NR-tree). But the bottleneck of the root of NR-tree is not solved completely. In addition, Ma et al propose peer-R-tree, which is similar with NR-tree [12]. Zhao et al. propose PR-tree, which is more suitable in multi-dimensional data queries [13]. Although all above approaches have higher efficiency, but the load balance problem of the tree-based P2P index network is inrooted. Therefore, the second paradigm of P2P spatial index networks using DHT-based structured P2P model is emerged recently. Tanin et al design a distributed MX-CIF Quad-tree, where the upper layer Quad-tree is mapped to the lower layer Chord P2P network [14]. They also introduce the fundamental minimum level, $f_{\min}$, to alleviate the root peer's burden and improve load balance of the system. But the locality of Chord network is weak, which increases the cost of geospatial query. Wang et al. propose an adaptive spatial peer-to-peernetwork (ASPEN) that extends CAN to preserve spatial locality information while also retaining many of the load balancing properties of DHT systems [15]. However, routing in CAN only uses neighboring information which takes long query time for a large scale P2P network. Ganesan et al use space-filling curves firstly to reduce multi-dimensional data into one dimension, and then utilize skip-graphs for efficient range queries [16]. Unfortunately, space filling curves do not always preserve locality and directionality. Kantere et al propose a spatial data storing and indexing P2P system: SPATIALP2P, which builds a new P2P network structure and preserves well locality and directionality of space [17]. But the proposed P2P network is relatively complex.

Inspired by these previous approaches, in this paper, we unite the above two paradigms and propose a new HPSIN, which combines distributed Quad-tree with DHT-based Chord network to make tradeoffs between query efficiency and system load balance, and achieve the overall high performance of the system. Although Tanin's approach also adopts Quad-tree and Chord network [14], our proposal is different in architecture, i.e. HPSIN is hybrid of Quad-tree and Chord network, whereas Tanin's is only mapping Quad-tree to Chord network. Moreover, based on opened queueing network, we also establish a theoretical model for HPSIN. There have been proposed some queueing network model for P2P file sharing networks [18–20]. However, as we know, our work is the first time that analyzes P2P spatial indexing network by queueing network theory. The contributions of our work are:

1) A new HPSIN is proposed, which combines the distributed Quad-tree with DHT-based Chord network to maintain both efficiency and load balance of the system.

2) A theoretical model based on opened queueing networks for HPSIN is established, which is simple but representative, and can capture the fundamental characteristics of the system.

3) Through numerical computing for the theoretical model, it is approved that there indeed exists an optimum point of tradeoff between efficiency and load balance, and can achieve the minimum overall query delay.

The rest of this paper is organized as follows: in Sect. 2, we describe the architecture of HPSIN. The theoretical queueing model of HPSIN is proposed and analyzed in detail in Sect. 3, and the results of numerical computing of the model are presented in Sect. 4. We finally conclude in Sect. 5.

2 HPSIN architecture

Geospatial data are multi-dimensional data with extent, which means each object is associated with more than one location. This character increases the complexity of spatial retrieval, which requires a set of intersection computations. In two-dimensional space, Geospatial queries are often executed by recursively subdividing the underlying space and then solving possibly simpler intersection problems. The recursive subdivision algorithm is the kernel of the common Quad-tree index and its variants. In Quad-tree, the underlying two-dimensional square-shaped space is recursively divided into four congruent square blocks until each block is contained in one of the data objects in its entirety or is not contained in any of the objects. Furthermore, each block can be uniquely identified by its centroid called control point. That's, each node in Quad-tree is equivalent to a control point which indexes the geospatial data within the corresponding block region. The Quad-tree cuts down the complexity of the intersection operation by enabling the pruning of certain objects or portions of objects from the query. In this paper, we also adopt Quad-tree and extend it to a distributed P2P environment for designing the HPSIN.

In P2P network, geospatial data objects are distributed and owned by all peers. According to Quad-tree algorithm, each object is indexed by one or multiple control points. In HPSIN, we use control point as the cluster formation decision of peers, that's, peers having the same control point fall into the same cluster. In each cluster, then, a header peer is elected to be a representation as the cluster. The header becomes an index node of Quad-tree, who will be assigned to join the upper layer cluster of corresponding control point and participate in the header election in this cluster. The above process is recursive until the root cluster header of the Quad-tree is decided.

However, in P2P environment, peers join and depart the network frequently, which may cause the P2P Quad-tree imbalanced and unstable. Therefore, an adaptive maintenance mechanism for P2P Quad-tree is needed. Supposing the P2P Quad-tree has been built initially. When a new peer wants to join a cluster of the Quad-tree, it should contact the corresponding header peer and become an ordinary member of the cluster directly. This process is simple and will not affect structure of the Quad-tree very much. While a header peer departs (either active or passive leaving) its own cluster, a new header should be elected by other peers in the cluster. If there is only one header peer in the cluster and it departs, the cluster is destroyed and the Quad-tree must be restructured partially. To alleviate the effect header peer departing and improve the stability of the tree, some policies should be considered, e.g. electing more powerful peer to be header, or deploying dedicated server as backup header, etc., whereas the detail of these policies is outside of scope of the paper and will be studied in future work.

For convenient, we denote $p_i(l)$ as the ith control point at level l in the Quad-tree, where $0 \leqslant l \leqslant l_{\max}$ and $l_{\max}$ is the maximum level of the tree. $\forall l$, $i=0,1,2,\ldots,4^l-1$, and $\forall p_j(l)$, $l < l_{\max}$, the four children control points of $p_j(l)$ is defined by $p_{4j}(l+1)$, $p_{4j+1}(l+1)$, $p_{4j+2}(l+1)$, $p_{4j+3}(l+1)$, which denote the four sub layer square blocks of $p_j(l)$ in clockwise respectively. Specially, the control point $p_0(0)$ is the root of tree. Fig. 1 is an example of control points in HPSIN with $0 \leqslant l \leqslant 2$, where the root is the centroid of the overall space and its children control points are $p_0(1)$, $p_1(1)$, $p_2(1)$, $p_3(1)$. Moreover, $p_0(1)$ has children of $p_0(2)$, $p_1(2)$, $p_2(2)$, $p_3(2)$, and $p_1(1)$ has children of $p_4(2)$, $p_5(2)$, $p_6(2)$, $p_7(2)$, etc.

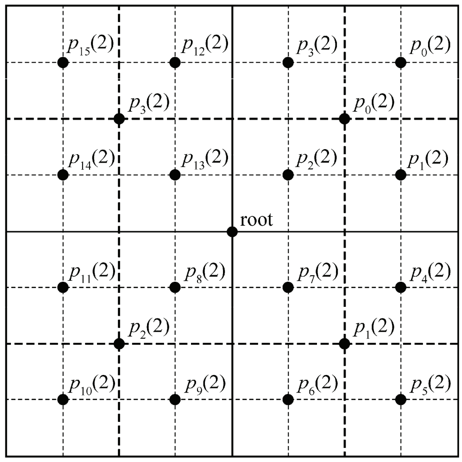

Fig. 1 An example of control points in HPSIN with $0 \leqslant l \leqslant 2$

Generally, a geospatial range query starts at the root of the Quad-tree and walks down along some branches of the tree. In a P2P environment of HPSIN, the tree traversal becomes a series of distributed cooperative operations of peers. This approach greatly alleviates the single server burden in traditional C/S based GIS. However, the upper layer peers of the Quad-tree (i.e. the root) will be the new bottleneck of the system. If the root is unavailable, the whole indexing network will be out of service. To handle this problem, in HPSIN, we introduce the start index level, l_s, which means that geospatial data objects can only be indexed at levels $l \geqslant l_s$, so all queries will also start at l_s. To locate the control points at level l_s in a distributed manner, we assign each control point $p_i(l_s)$ with a random identifier which is a mapping of the control point's coordinate through a consistent hash function (i.e. SHA-1). Then, using the identifier, every control point at level l_s joins the well-known DHT-based P2P network of Chord [5]. Therefore, instead of starting at the root, geospatial query starts at level l_s through Chord routing which is able to preserve scalability and system load balance. Fig. 2 is the architecture of HPSIN with $l_s = 1$. The reason we call HPSIN is hybrid is that, seen from the figure, the overall system is made up of two portions that are corresponding to the two different P2P geospatial data index paradigms mentioned above. The first portion $l = l_s$ is the DHT-based structured P2P network of Chord, and the second portion $l > l_s$ is the hierarchical distributed P2P Quad-tree. When $l_s=0$, the HPSIN is reduced to a distributed Quad-tree alone which has high efficiency of geospatial query but will cause hotspot at root. When $l_s=l_{\max}$, the HPSIN is reduced to a pure Chord indexing

P2P network which can improve the load balance of the system but will increase the query costs on the contrary. Taking an appropriate value of l_s, HPSIN can achieve a tradeoff between efficiency of query and load balance of the system, furthermore, enhance the overall performance of the system. We will discuss this topic in the following sections.

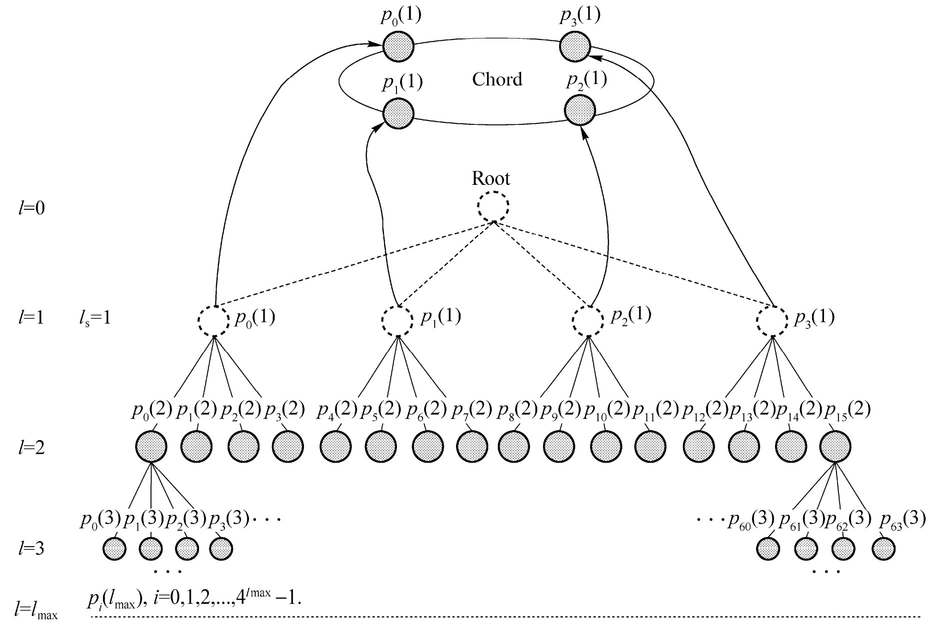

Fig. 2 Architecture of HPSIN with $l_s = 1$

3 Model and performance analysis

In order to understand the distinguishing characteristics of HPSIN, we seek a simple but representative theoretical analysis for the system. According to the architecture of HPSIN, usually, a geospatial query will travel through Chord network and P2P Quad-tree orderly. So, we can denote the overall delay of a query as D_A, which has

$$D_A = D_C + D_T \tag{1}$$

where D_C and D_T denote the delay in Chord network and P2P Quad-tree respectively. The delay in each P2P network is also made up of two parts: waiting delay before receiving service at peers (W_C or W_T) and network propagation delay (P_C or P_T) . We have

$$D_C = W_C + P_C \tag{2}$$

$$D_T = W_T + P_T \tag{3}$$

The propagation delay is arguably predictable. Because, for Chord network, the average routing hops of a query is about $\mathrm{lb}4^{l_s} = 2l_s$, so

$$P_C = 2l_s d \tag{4}$$

where d is the average propagation delay between two peers in physical network. Similarly, for Quad-tree, we have

$$P_T = (l_s - l_m)d \tag{5}$$

However, W_C and W_T are not easy to be estimated intuitionally. Therefore, we introduce queueing theory and model HPSIN as an opened queueing network to evaluate W_C and W_T [18]. For clarity, we divide the model into two partitions for Chord network and P2P Quad-tree respectively.

3.1 Chord network model

Generally, a geospatial query has a range. Assuming that the geospatial average query rectangle size is a fraction α $(0 < \alpha \leqslant 1)$ of the space in each dimension, then the number of control points at level l accessed by a query is given as $\max(4^l\alpha^2, 1)$, where the number of 1 can guarantee the query to be delivered to sub layer while $\alpha^2 4^l < 1$. Let λ be the geospatial query rate of a client, and the range of all queries is distributed uniform randomly in space, the average geospatial query rate for a control point at level l, $\lambda(l)$, is

$$\lambda(l) = \lambda \frac{\max(4^l\alpha^2, 1)}{4^l} \tag{6}$$

Because Chord network can preserve load balance among the peers, which is at level l_s with size of 4^{l_s} , the average geospatial query rate of each peer in Chord, λ_C , can be expressed as

$$\lambda_C = \lambda(l_s) \tag{7}$$

In addition, every geospatial query of λ_C will generate extra $\mathrm{lb}4^{l_s} = 2l_s$ routing query message in Chord. So the total query rate of each peer in Chord is $\lambda_C(2l_s+1)$, then the opened queueing network model for Chord can be shown as Fig. 3.

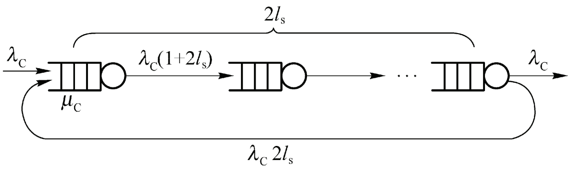

Fig. 3 Model for Chord network

In the model, the average routing hops for a query is about $2l_s$, and there are $2l_s$ series peers in Chord. We propose each peer of Chord network as M/M/1 processor. It has been shown that the M/M/1 queueing model is a good approximation in most of service system and can reveal the fundamental characters of the system through tackleable mathematical method. According to M/M/1 queueing theory, we get the utilization of a peer in Chord, ρ_C, as

$$\rho_C = \frac{\lambda_C(2l_s+1)}{\mu_C} \tag{8}$$

where μ_C denotes the service rate of a peer in Chord. Then, the average queueing length in the peer L_C is given by

$$L_C = \frac{\rho_C}{1-\rho_C} = \frac{\lambda\alpha^2(2l_s+1)}{\mu_C - \lambda\alpha^2(2l_s+1)} \tag{9}$$

The total queueing length in Chord, $\tilde{L}_C$, is

$$\tilde{L}_C = 2l_s L_C \tag{10}$$

Therefore, by applying Little's Law, the queueing and waiting delay in Chord network is given by

$$W_C = \frac{\tilde{L}_C}{\lambda_C} = \frac{2l_s(2l_s+1)}{\mu_C - \lambda\dfrac{\max(4^{l_s}\alpha^2,1)(2l_s+1)}{4^{l_s}}} \tag{11}$$

Combining Eq. (4) and Eq. (11), we get the average query delay D_C in Chord network.

3.2 P2P Quad-tree network model

Similarly, we can also model the P2P Quad-tree network as an opened queueing network and each peer of Quad-tree as M/M/1 processor which service rate is μ_T. In HPSIN, since the P2P Quad-tree starts at level l_s, the average query rate for each control point at level l ($l_s \leqslant l \leqslant l_{max}$) in Quad-tree, $\lambda_T(l)$, can be given by

$$\lambda_T(l) = \lambda(l) \tag{12}$$

Fig. 4 is a model for a peer at level l of the Quad-tree. According to the queueing theory, it is easy to obtain the waiting delay at level l of the Quad-tree, $W_T(l)$, as

$$W_T(l) = \frac{1}{\mu_T - \lambda_T(l)} = \frac{1}{\mu_T - \lambda\dfrac{\max(4^l\alpha^2,1)}{4^l}} \tag{13}$$

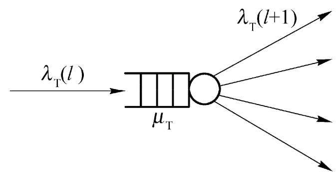

Fig. 4 Model for a peer at level l of the Quad-tree

Then, the total queueing and waiting delay in P2P Quad-tree network is

$$W_T = \sum_{l=l_s}^{l_{max}} W_T(l) \tag{14}$$

Combining Eq. (3) and Eq. (14), we also get the average query delay D_T in P2P Quad-tree network.

According to the above results, it can be proved that, with the increase of l_s, D_C is increased, but D_T is decreased contrarily. Hence, we propose interestingly there exits an appropriate value of l_s, which can make a tradeoff between efficiency and load balance, and achieve the minimum overall query delay D_A of the system. We will give the results of our model later.

4 Model results

In this section, we validate our queueing network model for HPSIN through numerical computing to have an insight of the system further. To obtain the numerical results without loss of generality, we choose the value of some system parameters as $\lambda = 1.0\ \mathrm{s}^{-1}$, $\alpha = 0.04$, $l_{max} = 12$, $\mu_C = 70.0\ \mathrm{s}^{-1}$, $\mu_T = 1.5\ \mathrm{s}^{-1}$, $d = 0.050\ \mathrm{s}$ and l_s varies from 0 to 12.

Fig. 5 shows the curves of the query delay versus the start index level l_s. From this Figure, it can be seen that D_C is increased with the increase of l_s, whereas D_T is decreased with the increase of l_s, which is consistent with the theoretical model. For the curve of overall delay of query D_A, there indeed exists a point of l_s, $l_s = 5$ in Fig. 5, where the value of D_A is the minimum. We call this point of l_s the optimum l_s, and denote it as l_s(opt).

In addition, we evaluate relationship between l_s(opt) and query rate λ with different α. As shown in Fig. 6, when α is smaller, i.e. $\alpha = 0.01$, l_s(opt) is no change with increase of λ; when α is greater, i.e. $\alpha = 0.9$, l_s(opt) is increased with the increase of λ. This phenomenon enlightens us that when the query rate λ is increased, we should enlarge dynamically l_s to the value of l_s(opt) to achieve the minimum delay of query.

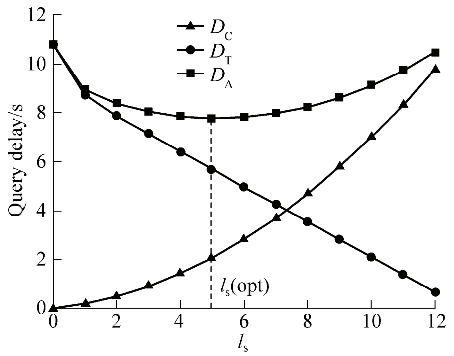

Fig. 5 Query delay vs. l_s

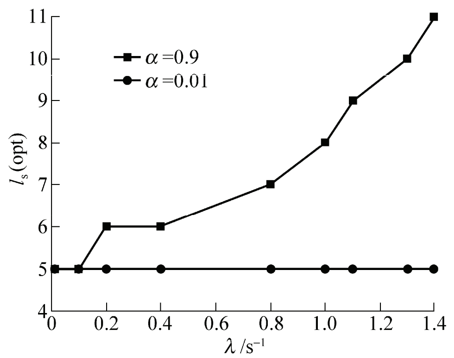

Fig. 6 l_s(opt) vs. λ with different α

Finally, we plot the curves of l_s(opt) vs. l_{max} with different α, where $\lambda = 1.0\ s^{-1}$ in Fig. 7. Since the size of HPSIN grows exponentially with l_{max}, Fig. 7 reflects the relationship between l_s(opt) and the scale of indexing network. For α=0.01, when $l_m \leqslant 5$, l_s(opt) is the equal to l_{max}; when $l_{max} > 5$, l_s(opt) is constantly equal to 5 which is smaller than l_{max}. This means when the scale of HPSIN is small, a pure structured P2P network of Chord should be adopted; while the scale becomes large, HPSIN should transform from the pure structured network to a hybrid P2P network to preserve high performance. Similarly, the situation of $\alpha = 0.9$ is as same as $\alpha = 0.01$, except that the transformation point of l_{max} increases to 8. The results of both Fig. 6 and Fig. 7 suggest that we should configure a proper value of l_s according to different network scale and query rate, so as to make HPSIN have capacity of adaptability in different P2P application environments.

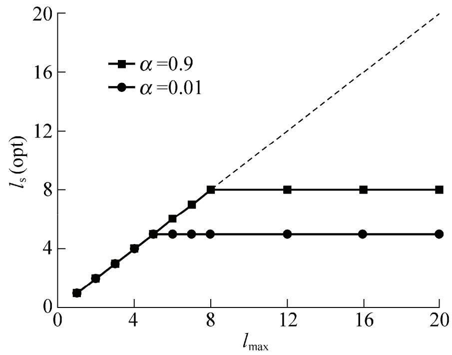

Fig. 7 l_s(opt) vs. l_{max} with different α

5 Conclusions

In this paper, combining distributed Quad-tree with Chord network, we propose a new HPSIN as well as its theoretical model based on opened queueing network. The theoretical analysis and numerical computing results show that there exists an optimum value of l_s which can obtain the minimum total query delay through making tradeoff between efficiency and load balance. There are several issues can be studied in future work, i.e. the M/M/1 model of peers can be extended to more general model, detailed policies to maintain HPSIN when peers join and leave dynamically should be studied, and the optimum strategy for choosing l_s(opt) online may be proposed.

Acknowledgements

This work was supported by the Hi-Tech Research and Development Program of China (2009AA12Z219), the National Natural Science Foundation of China (40801149), the Jiangsu Province Bureau of Surveying and Mappings Research Fund (JSCHKY200810), Visiting Scholar Foundation of Key Lab. In University, the NJUPT Research Fund (NY206009).

References

1. Finkel R A, Bentley J L. Quad trees: a data structure for retrieval on composite keys. Acta Informatica, 1974, 4(1): 1–9
2. Skvortsov A V. Algorithms for improving the quality of R-trees. Russian Physics Journal, 2001,44(6): 588–595
3. Ripeanu M. Peer-to-peer architecture case study: Gnutella network. Technical report, TR-2001–26. Chicago, IL, USA: University of Chicago, 2001
4. Clarke I, Sandberg O, Wiley B, et al. Freenet: a distributed anonymous information storage and retrieval system. Designing Privacy Enhancing Technologies: Proceedings of the ICSI Workshop on Design Issues in Anonymity and Unobservability, Jul 25–26, 2000, Berkeley, CA, USA. LNCS 2009. Berlin,Germany: Springer-Verlag, 2000: 44–66

5. Stoica I, Morris R, Karger D, et al. Chord: a scalable peer-to-peer lookup service for internet applications. Proceedings of the International Conference of the Special Interest Group on Data Communication (SIGCOMM'01), Aug 27−31, 2001, San Diego, CA, USA. New York, NY, USA: ACM, 2001: 149−160
6. Ratnasamy S, Francis P, Handley M, et al. a scalable content-addressable network. Proceedings of the International Conference of the Special Interest Group on Data Communication (SIGCOMM'01), Aug 27−31, 2001, San Diego, CA, USA. New York, NY, USA: ACM, 2001: 161−172
7. Rowstron A, Druschel P. Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems. Proceedings of the 18th IFIP/ACM International Conference on Distributed Systems Platforms (Middleware'01), Nov 12−16, 2001, Heidelberg, Germany. LNCS 2218. Berlin, Germany: Springer-Verlag, 2001: 329−350
8. Banerjee S, Bhattacharjee B, Kommareddy C. Scalable application layer multicast. Proceedings of the International Conference of the Special Interest Group on Data Communication (SIGCOMM'02), Aug 19−23, 2002, Pittsburgh, PA, USA. New York, NY, USA: ACM, 2002: 205−217
9. Tran D A, Hua K A, Do T T. ZIGZAG: an efficient peer-to-peer scheme for media streaming. Proceedings of IEEE the 22nd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'03): Vol 2, Mar 30−Apr 3, 2003, San Francisco, CA, USA. Piscataway, NJ, USA: IEEE, 2003: 1283−1292
10. Mondal A, Yi L F, Kitsuregawa M. P2PR-tree: an R-tree based spatial index for peer-to-peer environments. Proceedings of the International Workshop on Peer-to-Peer Computing and Databases (P2P&DB'04), Mar 14, 2004, Heraklion, Greece. LNCS 3268. Berlin, Germany: Springer-Verlag, 2004: 516−525
11. Liu B, Lee W C, Lee D L. Supporting complex multi-dimensional queries in P2P systems. Proceedings of the 25th International Conference on Distributed Computing Systems (ICDCS'05), Jun 5−10, 2005, Columbus, OH, USA. Piscataway, NJ, USA: IEEE, 2005: 155−164
12. Ma X J, Liu C, Xie K Q, et al. A research on global spatial data directory in peer-to-peer networks. Geography and Geo-Information Science, 2006, 22(3): 22−25 (in Chinese)
13. Zhao F F, Zhang J. Analysis on multi-scale spatial data index structure. Engineering of Surveying and Mapping, 2008, 17(2): 26−29 (in Chinese)
14. Tanin E, Harwood A, Samet H. Using a distributed QuadTree index in peer-to-peer networks. The International Journal on Very Large Data Bases, 2007, 16 (2):165−178
15. Wang H J, Zimmermann R, Ku W S. ASPEN: an adaptive spatial peer-to-peer network. Proceedings of the 13th ACM International Workshop on Geographic Information Systems (ACM-GIS'05), Nov 4−5, 2005, Bremen, Germany. New York, NY, USA: ACM, 2005: 230−239
16. Ganesan P, Yang B, Garcia-Molina H. One torus to rule them all: multidimensional queries in P2P systems. Proceedings of the 7th International Workshop on the World Wide Web and Databases (WebDB'04), Jun 1−6, 2004, Paris, France. New York, NY, USA: ACM, 2004: 19−24
17. Kantere V, Skiadopoulos S, Sellis T. Storing and indexing spatial data in P2P systems. IEEE Transactions on Knowledge and Data Engineering, 2009, 21(2): 287−300
18. Bloch G, Greiner S, De Meer H, et al. Queueing networks and markov chains. New York, NY, USA: John Wiley and Sons, 2006
19. Ge Z H, Figueiredo D R, Jaiswal S, et al. Modeling peer-peer file sharing systems. Proceedings of IEEE the 22nd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'03): Vol 3, Mar 30−Apr 3, 2003, San Francisco, CA, USA. Piscataway, NJ, USA: IEEE, 2003: 2188−2198
20. Ramachandran K K, Sikdar B. An analytic framework for modeling peer to peer networks. Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'05): Vol 3, Mar 13−17, 2005, Miami, FL, USA. Piscataway, NJ, USA: IEEE, 2005: 2159−2169

(Editor: ZHANG Ying)