AI for Peer-to-Peer Systems

Chaokun Wang
Dept. of Computer Science and Engineering, Harbin Institute of Technology, China
wangchaokun@0451.com

Jianzhong Li
Dept. of Computer Science and Engineering, Harbin Institute of Technology, China
lijz@mail.banner.com.cn

Shengfei Shi
Dept. of Computer Science and Engineering, Harbin Institute of Technology, China
shengfei@0451.com

Abstract

Peer-to-Peer systems have become a focus of distributed computing these years. Document discovery is a key problem in this field. In this paper, abstract indices are proposed for content-based information retrieval in Peer-to-Peer systems, especially for multimedia documents. Abstract information is extracted from the shared documents of each peer, then the similar documents on other peers can be retrieved based on content via abstract indices in both centralized P2P systems and distributed P2P systems. Abstract indices (AI) make it possible to search fewer peers but get more similar documents. Experiments show AI works well in simulated P2P systems.

Keywords

Peer-to-Peer Systems, Content-Based Information Retrieval, Abstract, Abstract Indices, Similarity Search, Document Discovery, Multimedia Documents

1 Introduction

Peer-to-peer (P2P) is a dramatically increasing research field in these years. As the basis of the Internet, now P2P becomes the focus of people. P2P systems allow users to exchange files, communicate instantly, store documents safely and effectively, and so on. There is a very large collection of documents in a P2P system. So it is an important problem to retrieve documents based on content in this system. Our aim is to search fewer peers but get more results in content-based information retrieval, especially the multimedia documents.

There are two kinds of peer-to-peer systems, each with their own advantages and disadvantages. The first is the centralized P2P system. One or more nodes exist in a centralized P2P system acting as the center of the system. This kind of nodes is named the coordinator, supernode, or other words. Indices of the whole system may be established on the coordinator. To process a query, the coordinator has to be used to locate the appropriate peers. The hybrid P2P system is only a variation of the centralized P2P system.

The second is the distributed P2P system, or the pure P2P system. In this kind of system, local index may be created on each peer. The distributed P2P system works by flooding the network (or a subset of it) to process each query submitted from any peer.

In this paper the abstract indices (AI) are proposed. Abstract is the summary of the shared documents of a peer. Abstract index is the index established on the abstracts of all peers in a system. The content-based information retrieval, especially to multimedia documents, can be solved via AI. AI works well in the simulated experiments for the both kinds of P2P systems.

The rest of this paper is organized as follows. In next section the related work is reviewed. The related work is reviewed. In section 3 the problem of document discovery in P2P systems is discussed. Abstract Information of a Peer is presented in section 4. Then abstract indices for centralized and distributed Peer-to-Peer Systems are respectively discussed in section 5 and section 6. And the algorithms for abstract indices are presented in section 7. The results of the simulated experiments are given in section 8.

2 Related Work

Many P2P systems have been proposed and developed [UG, UF, UB, US, UL]. But they all lack the support for content-based information retrieval, especially to multimedia documents. That is these systems work on document identifiers, rather than on the content of the documents. [CN02] gives a method for the content-based text information retrieval in P2P environments. This work can not extend to other media, e.g. image.

A distributed lookup protocol can be found in [SMK01] to solve the locating problem in peer-to-peer applications. [CG02] proposes a distributed-index mechanism, named route indices. The routing indices are maintained on each peer. It gives a "direction" towards the document, rather than its actual location. Unfortunately, RI only works for distributed P2P systems. [YG02] focuses on search techniques that do not use indexes, although it also studies one type of local area index.

[RD01] describes a kind of storage management and caching in a P2P environment. [KNO02] focuses on the application of OLAP in P2P environments. [GHI01] discusses the relation between database and P2P.

The problem of database selection, i.e. selecting the best databases to which to send a query, is studied in [GGT00, CGJ02]. However, the number of peers in a P2P system is more than that of databases in these papers, and the autonomy of each peer is higher than that of each database in these multi-database environments.

Some recent work focuses on the evaluation of P2P systems. [YG01] develops a probabilistic model and an analytic model for “hybrid” P2P systems, but it doesn’t consider other kinds of P2P systems. And the models cannot deal with content-based information retrieval. An evaluation and description of the present state of Gnutella is given in [Clip].

3 The problem of document discovery in P2P systems

3.1 Overview of centralized P2P systems

In a centralized P2P system, there are always one or several peers acting as the center of the system. The meta-data or feature of each document in the system is sent to the central peer (perhaps named coordinator, supernode, or other words). When a user submits a query, the query or the feature of the query should be sent to the coordinator(s), where the query process is done.

There are some drawbacks in the centralized P2P systems. Firstly the autonomy of peers is disobeyed. Each peer has to submit the meta-data or feature of each shared document to the coordinator. Secondly the network load is high. The transfer of the features of all documents is an onerous task. Thirdly the central index is too large and is vulnerable to attack. Finally there is no efficient way to deal with the content-based multimedia information retrieval in this kind of system. One simple method is to retrieve document in the coordinator, but it needs all features should be stored on the coordinator. It not obeys the autonomy of the peer, because perhaps a peer doesn't want the coordinator know what is stored on itself in detail. As far as we know, there is not formal publication about this problem.

3.2 Overview of Distributed-Search P2P Systems

In a distributed P2P system, there are so many peers having the same degree. To discover some certain documents, the existing methods should search all neighbors of the querying peer and the messages flood the system. And the summation of computation on each peer is very enormous. There is no formal publication about an efficient and effective method for multimedia document retrieval based on content.

3.3 The problem of document discovery

The problem of document discovery in P2P systems is to minimize the whole computation of the whole system but get more satisfying results.

The problem can be formalized to the following optimization problem.

min SC(P), NL(P)

sub to. SR > n

Where P is a P2P system, SC(P) is the summation of computation in the whole system, NL(P) is the whole network load, SR is the number of documents as the final results satisfying the query, and n is a positive integer.

In a P2P system, there is usually a very large collection of documents. And there are so many documents satisfying the user’s demands that a part of them is enough to the user. Then the exactly K-Nearest-Neighbors problem in a P2P system is not important, and it is not considered in this paper.

4 Abstract Information of a Peer

In the rest of this paper, the abstract of the shared documents of a peer is always named the abstract of the peer. The abstract of a peer is discussed in this section. Abstract Indices of P2P systems will be discussed in the following two sections.

4.1 What is the abstract information of a peer?

For a set of pictures about cars, the word "car" can be used to describe this set. This kind of words, i.e. car, is usually named "meta-data". In content-based information retrieval, "meta-data" can be used to label these pictures, but it is better to retrieve the similar objects to the query based on content instead of "meta-data". In Peer-to-Peer environments, each peer can be considered as a general object, so it is a good idea to find some features to describe it like the effect of "meta-data".

The abstract of a peer is the "meta-data" of it, but it is based on the content of the shared documents on this peer. The abstract of a peer is the information that can characterize the shared documents of this peer. It can be thought as the thumbnail of the shared documents on the peer.

There exists the abstract of a peer in P2P systems. A P2P system is usually constructed for a special group. Different communities can have different P2P systems. In a community having a same interesting, such as a music community, each person shares the music collection of his personal computer, i.e. a peer of the P2P system. If he likes classical music, there perhaps lots of classical music in his shared collection. So for each peer, there may be one or several features that can represent the shared documents on it. The methods of creating abstract of a peer will be discussed in the next subsection.

The abstract of a peer can be applied to document discovery in P2P systems. Given a query, some peers, which seem that a lot of documents similar to the query are stored in, can be selected via the abstracts of all peers in a P2P system. Then fewer peers will be searched but more results will be returned.

4.2 How to create the abstract of a peer?

There are two steps to creating the abstract of a peer. The first is the selection of the set of attributes used to represent the document, and the second is the production of the abstract of the documents of a peer.

The selection of the attributes set is very important. These attributes are expected enable characterizing a document. For example, pitch, rhythm, timbre and dynamics can be used to represent a piece of music [BC02]. The selection of the attributes set of multimedia documents is very hard because the meaning of multimedia documents can be varying to different persons.

There are many methods to create the abstract of a peer. For example, to the set of shared documents of a peer, some clustering methods can be used to cluster the features of these documents based on their content. Then these clustering centers of all clusters can be used as the abstract of the peer.

Other methods, such as classification, statistical calculation can be used to create the abstract information. The creation of the abstract of a peer is very difficult problem, but it is also very interesting. It is still an open question to create the proper abstract of a peer.

In this paper, a statistical method is adopted to create the abstract of a peer. Suppose that a P2P system store a certain kind of document, and X_1, X_2, …, $X_{,n}$ is the *n* random variables corresponding to the *n* features of this kind of document, then each document in this system can be represented as a vector (x_1, x_2, …, x_n). The *n*+1-tuple (y_1, y_2, …, y_n, y_{n+1}) is named the abstract of a peer where

$$y_i = E(X_i), \qquad i = 1, \cdots, n,$$

$$y_{n+1} = E\left(\sqrt{diag(C)}\right),$$

C is the covariance matrix of X_1, X_2, …, X_n.

For example, in an image P2P system, a 16-dimension histogram is selected as the feature of the image. Then each image is represented as a 16 dimensions vector. The value of each dimension is normalized to a number between 0 and 1. X_i is the random variable corresponding to the i-th dimension. A 17-tuple (y_1, y_2, …, y_{16}, y_{17}) is computed from the set of shared images of a peer as the abstract, where y_i is the mean of X_i, y_{17} is the mean of the standard variances of all X_i (i=1, …, 16). The abstract of a peer can be plotted as a point in the 17-dimension space.

4.3 Others

There is update issue about abstract of a peer. When the shared documents of a peer change, e.g., some documents are added into the shared collection, the abstract of this peer should be updated. The degree of update of abstract varies. For example, if a clustering algorithm is used as the abstract-creating method, when a single document is added, it is not necessary to alter all central points of all clusters in the abstract, perhaps one central point is enough. When many documents are added, perhaps several central points should be altered, and even some new central points are created.

5 AI for centralized Peer-to-Peer Systems

It is very interesting to organize the abstracts of all peers in a P2P system. A kind of index of abstracts, named abstract indices (AI), is introduced in this paper. In this section, AI in centralized P2P systems is discussed. And AI in distributed P2P systems will be discussed in the next section.

The objective of AI is to search fewer peers but get more results, i.e. minimize the computation of the whole system but maximize the efficiency of the document discovery.

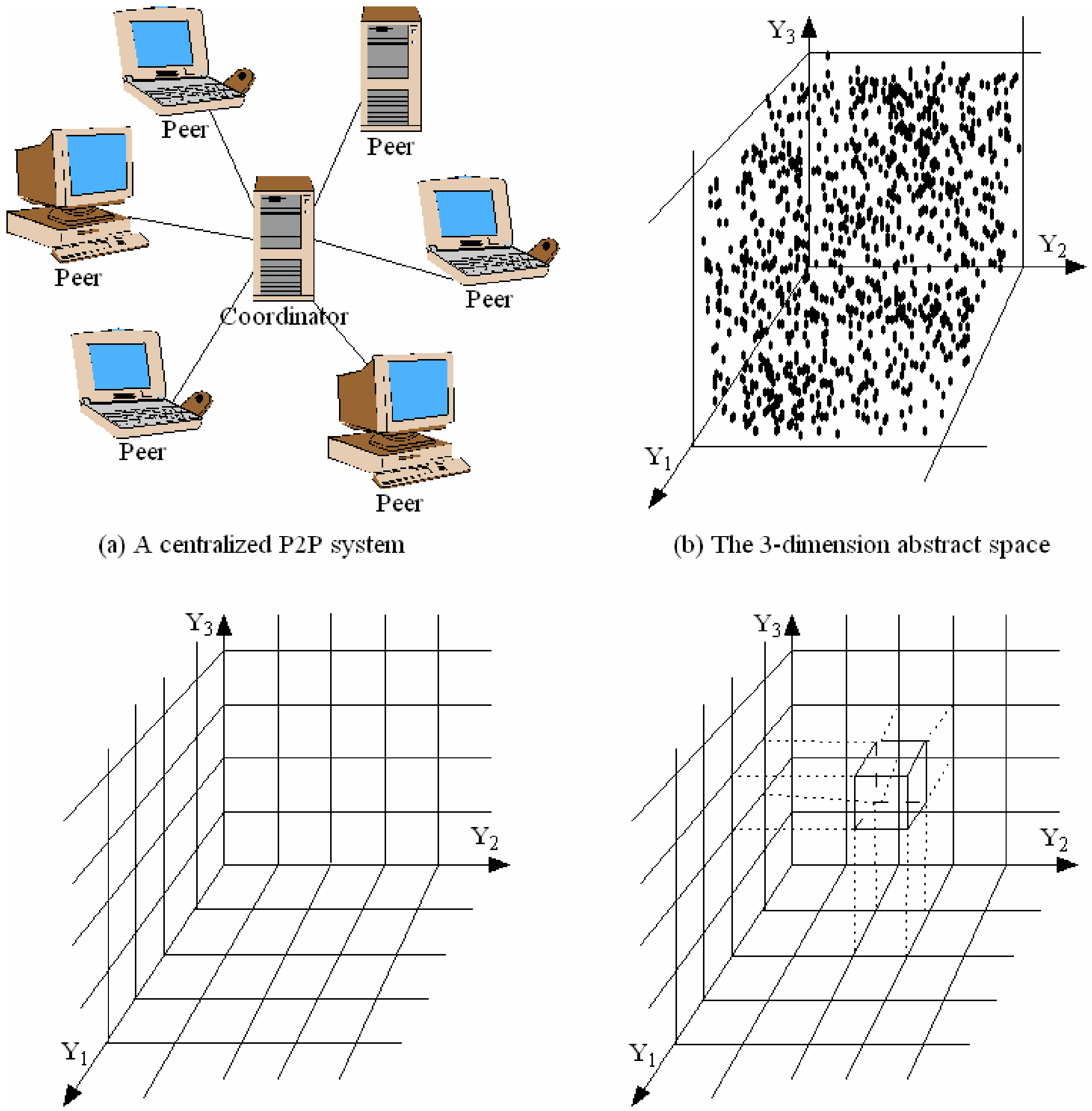
The AI in centralized P2P systems allows a coordinator to select the "best" peers to send a query to. An AI in a centralized P2P system is a data structure that, given a query, returns a list of peers, which a lot of documents similar to the query may be stored on.

In this section and the next, a simple P2P environment of using AI is given. This environment comprises of 1000 peers, 500 objects are stored in each peer. The selected attributes of object are X_1 and X_2. Then each object can be represented as a 2-tuple (x_1, x_2), and x_1 and x_2 can be normalized into [0, 1]. The statistical abstract-creating method in section 4.2 is used in this system, i.e., the statistical feature values of (x_1, x_2) of all objects in a peer can be used to characterize this peer. Then each peer can be represented as a

3-tuple (y_1, y_2, y_3), where y_1 is the mean of X_1, y_2 the mean of X_2, and y_3 the mean of the standard deviations of X_1 and X_2. This 3-tuple is named the abstract of the peer. Each abstract can be plotted as a point in the 3-dimension space where the minimum and the maximum value of each dimension is respectively 0 and 1.

5.1 Creating Abstract Indices

In this subsection, an AI for a centralized P2P system is created. The P2P system is established on the frontal simple P2P environment. For simplicity, this system has only one coordinator. The system is depicted in the figure 1-a.



(a) A centralized P2P system

(b) The 3-dimension abstract space

(c) The partition of the abstract space

(d) One subspace of the abstract space

Figure 1. AI in a centralized P2P system

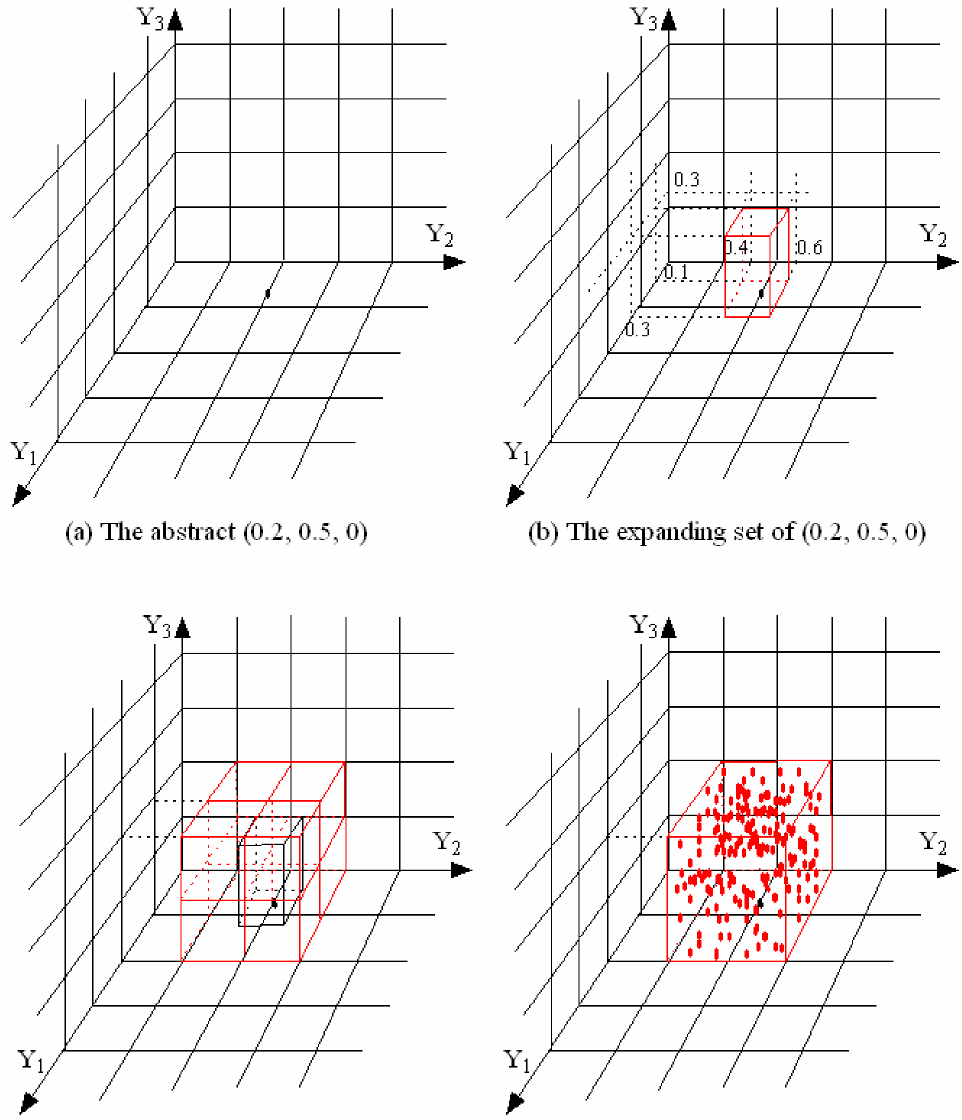
The abstracts of the 1000 peers are mapped to 1000 points in the 3-dimension abstract space (figure 1-b). The abstract space can be partitioned into several subspaces. Given the partitioning sequences of each dimensions is (0, 0.25, 0.5, 0.75, 1), the space is partitioned into 4^3 subspaces (figure 1-c). One of the subspaces is illustrated in Figure 1-d. Each abstract is mapped into a subspace, and each subspace corresponds to a set of abstracts of peers, i.e., a set of peers, which are named the peers of the subspace.

The organization of the abstract spaces is named abstract index. In this paper, an AI is the partition of the abstract space, i.e., the data structure (and corresponding algorithms) of storing the mapping from abstract point set to the exponential set of the abstract subspace set and storing the peers of each subspace.

AI is not just hashing. Hashing is the mapping from a point to a bucket, but the mapping in AI "maps" a point to several subspaces (buckets). AI is more general than hashing.

5.2 Using Abstract Indices

In centralized P2P systems, AI is usually saved on the coordinator. By using AI, a set of peers will be selected as "the likely-destination peers", i.e. peers where perhaps a lot of documents similar to the query are stored. In this subsection, an example is given to describe how to use abstract indices during the query process.



(a) The abstract (0.2, 0.5, 0)   (b) The expanding set of (0.2, 0.5, 0)

(c) The minimal overlay of the expanding set   (d) The union of the peers of these subspaces in the minimum overlay

Figure 2. Using AI in a centralized P2P system

When a user puts forward a query from any peer, this query can be represented as a 2-tuple, such as (0.2, 0.5). It can be thought as the abstract of a virtual peer that only this query is stored on. Then the abstract of the virtual peer is (0.2, 0.5, 0). This abstract will be sent to the coordinator and mapped into the 3-dimension space (figure 2-a).

Given the expanding factors in Y_1, Y_2 and Y_3 are respectively 0.1, 0.1 and 0.3, and then the abstract of the virtual peer can be expanded to an area in the abstract space. This area $\{(y_1, y_2, y_3) \mid 0.1 \le y_1 \le 0.3, 0.4 \le y_2 \le 0.6, 0 \le y_3 \le 0.3\}$ is named the expanding set of the abstract (0.2, 0.5, 0) (figure 2-b).

A set of subspaces can be used to overlay the expanding set of the abstract. The smallest such set is named the minimum overlay of the expanding set. The minimum overlay of the expanding set of (0.2, 0.5, 0) is the set of subspaces $S_{1,2,1}$, $S_{1,2,2}$, $S_{1,3,1}$, $S_{1,3,2}$, $S_{2,2,1}$, $S_{2,2,2}$, $S_{2,3,1}$, and $S_{2,3,2}$ (figure 2-c). The union of the peers of these subspaces in the minimum overlay is the set of the likely-destination peers to the query (figure 2-d).

5.3 Maintaining Abstract Indices

It is very important to maintain abstract indices for the validity of AI. When a peer connects into a centralized P2P system, its abstract should be added to the coordinator. When a peer disconnects from this system, its abstract should be deleted from the coordinator. When the abstract of a peer is altered, such as some documents added, the change should propagate to AI. Usually batch update periodically and instant update are two policies in these cases. The cost of the former is less than that of the latter, but the effect of the latter is better than that of the former.

AI should be updated if the partitioning sequence of any dimension is altered. For the same abstract space, different partitioning points, e.g. the number or the values of them, can obtain different AI. So at the same expanding factors, the set of the likely-destination peers can still be refined if the partitioning sequences are tuned.

6 AI for distributed Peer-to-Peer Systems

The abstracts of all peers in a distributed P2P system comprise the abstract indices of the system. And the creation and maintenance of AI are equal to those of abstracts. However, in distributed peer-to-peer systems, AI can still be used to search fewer peers but get more results satisfying the user's query.

Without abstract mechanisms, a query is sent to all its neighbors to get the proper results. Each neighbor has to compare the query and all its shared documents. With AI, peers concerned with the query process are classified as reached peers and searched peers. A reached peer is the peer that receives the abstract of the virtual peer. A searched peer is the peer whose abstract belongs to the expanding set of the abstract of the virtual peer. Searched peers must be reached, but reached peers perhaps are not searched. Only the searched peers compare the query and its shared documents.

For example, given a distributed P2P system that is established on the frontal simple P2P environment, AI is created and cycles are not concerned. A part of the system is depicted in the figure 3-a. The user's query and the expanding factors in Y_1, Y_2 and Y_3 remain those in section 5.2. The query process can be described as follows. A user gives a query (0.2, 0.5) in P_1, and the abstract (0.2, 0.5, 0) of the virtual peer is extracted from it. P_1 finds that its abstract does not belong to the expanding set of (0.2, 0.5, 0), and then sends (0.2, 0.5, 0) to P_2, P_3 and P_4. P_2 finds its abstract belongs to the expanding set, then compares the query with its shared documents and sends the query to P_5, P_6 and P_7. The rest follows the rules. In figure 3-b, "S" denotes the searched peer, and "R" denotes the reached peer. It is obvious that the computation of the whole system goes down. And enough satisfying documents can also be gotten from the experimental results in section 8.

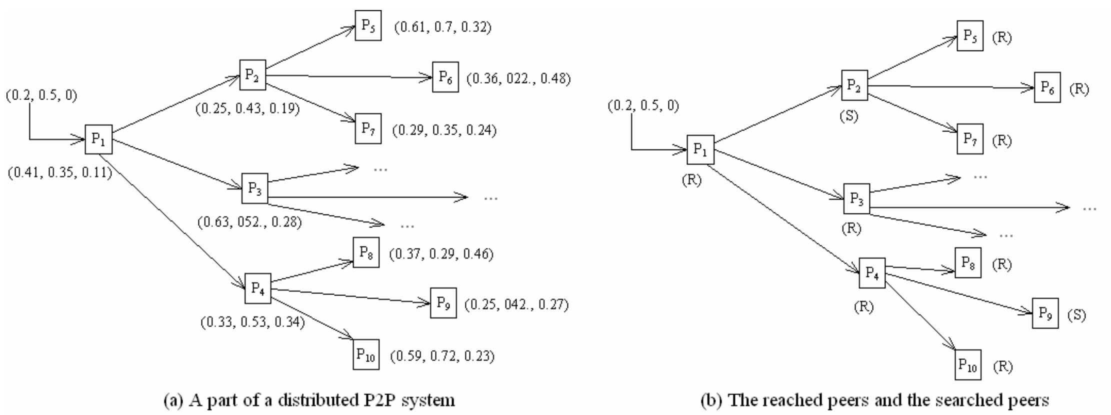


Figure 3. AI in a distributed P2P system

7 Algorithms for Abstract Indices

In this section, some algorithms about abstract indices are proposed. The algorithms for creating abstract of peer can be various, so it is not described here.

In P2P systems each peer is self-existent. It is very common that many copies of a multimedia file stored in different peers. So it is very important to filter out the replica in the results returned to users. There is a simple and effective method to filter out the repetition in the results. The multimedia files with the same content must have the same size and usually have the same date-time, therefore the size or the date-time of multimedia files with the equal similar degree can be compared to judge whether these files are repeated. Usually the size is enough.

7.1. Algorithms for creating AI in centralized P2P systems

AI Creation Algorithm in centralized P2P systems

Input: Abstracts of all peers in the system, partitioning sequences of each dimensions

Output: AI in the coordinator

1) Partition the abstract space into subspaces based on the partitioning sequences;

2) For each subspace

record the peers of this subspace;

7.2. Algorithms for answering a query in centralized P2P systems

Input: a query, expanding factors, AI, similarity conditions

Output: the final results

1) Compute the abstract of the virtual peer that comprises only the query;

2) Compute the expanding set of the abstract using the expanding factors;

3) Compute the minimal overlay of the expanding set;

4) Compute the union of the peers of subspaces which belongs to the minimal overlay, i.e. the set of likely-destination peers;

5) For each likely-destination peer

Retrieve the similar documents similar to the query and return the documents satisfied the similarity conditions;

6) Filter the results and produce the final results;

7.3. Algorithms for answering a query in distributed P2P systems

Input: a query, expanding factors, AI, similarity conditions

Output: the final results

1) Compute the abstract of the virtual peer that comprises only the query;

2) For each peer reached,

If the peer is searched, i.e. the abstract of this peer belongs to the expanding set of the query

Retrieve the similar documents similar to the query and return the documents satisfied the similarity conditions;

3) Filter the results and produce the final results;

8 Experiments

8.1 Experimental Setup

AI can be used for the content-based information retrieval in P2P systems, especially to multimedia documents. A simulated multimedia document P2P system, e.g. an image P2P system, is constructed in this paper. Each image is represented as a 3-tuple, i.e. (R, G, B), where each element of the tuple is a number between 0 and 1 and represents the degree of the corresponding color. The two images which Euclidean distance is within a certain value are thought as the similar images.

The simulated image P2P system contains 10000 peers. Each peer has some simulated images, i.e. 3-tuples, from 29 to 1000. The total number of images in the system is 2,805,028. Each dimension of the abstract of a peer is produced randomly under the normal distribution.

8.2 Experiments in the centralized P2P system

In the simulated centralized P2P system, only one coordinator exists. After a user submits an image query from any peer, the abstract of the virtual peer that consists of only the query image is extracted and sent to the coordinator. The coordinator retrieves the likely-destination peers from AI based on the abstract, and then sends the query to these peers to get detail results.

8.2.1 Some measurements

In this subsection, some measures are defined. The ratio of the number of the likely-destination peers to the number of all peers is named the peer-ratio (PR), which expresses the percentage of the likely-destination peers. Given a value of offset (In this paper, two images is said similar each other if the distance of their features is less than the value of offset.), the ratio of the number of the documents similar to the query in the results to the number of all documents similar to the query in the system is named the hit-ratio (HR). The ratio of HR to PR is named the accelerating-ratio (AR), which expresses the efficiency of the accelerating algorithm.

8.2.2. Results of the centralized P2P system

When the expanding factors in Y_1, Y_2, Y_3 and Y_4 are respectively 0.18, 0.18, 0.18 and 0.15, the results of retrieval by AI are illustrated in figure 4 and figure 5. They are the average

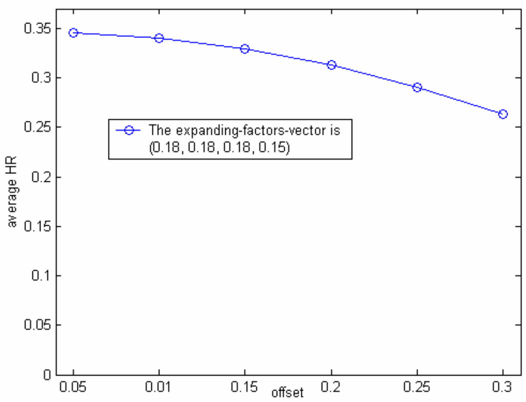


Figure 4. The average HR in the centralized P2P system

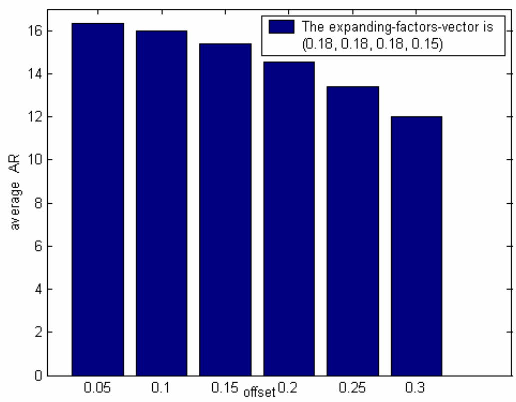


Figure 5. The average AR in the centralized P2P system

values on 512 test points. When the expanding factors are other values, the trends remain.

8.3 Experiments in the distributed P2P system

In a distributed P2P system, the number of the neighbors of a peer is named the width of the system. The number of the hops a query can propagate is named the depth of the system. In the simulated distributed P2P system, the width of the system is 5, and the depth of the system is 7. After a peer receives the abstract of the virtual peer that consists of only the query image, it judges whether it is the searched peer, i.e. whether its abstract belongs to the expanding set of the query abstract. If this peer is a searched peer, it searches its shared documents to retrieve the documents similar to the query. And this peer will send the query abstract to its neighbors until the hop of the message is more than the depth of the system.

8.3.1 Some measurements

In the distributed P2P systems, PR is the ratio of the number of searched peers to that of the reached ones, HR is the ratio of the number of the number of the documents similar to the query in the searched peers to the number of all documents similar to the query in the reached peers, and AR is still the ratio of HR to PR

Besides these measurements, the max-hit-ratio (MHR) is also a measure of distributed P2P systems. The max-hit-ratio is the ratio of the maximal number of documents similar to the query image in the results to the number of all documents similar to the query. The max-hit-ratio is determined by the topology of a P2P system, and is not the problem that

AI tries to solve.

8.3.2. Results of the distributed P2P system

In the experiment of the distributed P2P system, the expanding factors are the same in subsection 8.2.2. The value of offset is 0.1. The results by AI in different test points are listed in table 1. It shows that AI also works well in the distributed P2P system.

query point	HR	PR	AR	MHR
(0.20, 0.50, 0.70)	0.0063	0.3141	50.1125	0.5060
(0.30, 0.40, 0.60)	0.0075	0.2575	34.2383	0.4757

Table 1. The results of AI in a distributed P2P system

The experiments on more test points are working in progress.

9 Conclusion and Future Work

The contributions of this paper are: (1) we introduce abstract of a peer to summary the shared documents of a peer and to characterize the peer; (2) we introduce Abstract Indices, an efficient way for content-based information retrieval in P2P systems; (3) we present the two kinds of AI for both distributed and centralized P2P systems. Experiments show that AI work well in the simulated experiments for the both kinds of P2P systems.

More works about the experiments and performance analysis of AI will be done in the future.

10 Acknowledgments

This material is based on work supported in part by National Science Founds of China (69873014) and National 863 Plan of China (2001AA415410). And we thank the member of database laboratory at HIT.

References

[BC02] D. Byrd and T. Crawford. Problems of music information retrieval in the real world. Information Processing and Management, 38 (2002), pp249~272.

[CGJ02] J. Conrad, X. Guo, P. Jackson, and M. Meziou. Database Selection Using Actual Physical and Acquired Logical Collection Resources in a Massive Domain-specific Operational Environment, VLDB 2002

[Clip] Clip2.com, world-wide web: http://dss.clip2.com/gnutella.html. Gnutella: To the Bandwidth

and Beyond.

[CN02] F. Cuenca-Acuna and T. Nguyen. Text-Based Content Search and Retrieval in ad hoc P2P Communities. In proceedings of The International Workshop on Peer-to-Peer Computing, May 2002.

[GHI01] S. Gribble, A. Halevy, Z. Ives, M. Rodrig, and D. Suciu D. What Can Databases do for [GGT00] L. Gravano, H. Garcia-Molina, and A. Tomasic. Gloss: Text-source discovery over the internet. ACM Transactions on Database Systems (TODS'2000), 2000. Peer-to-Peer?. WebDB 2001

[KNO02] P. Kalnis, W. Ng, B. Ooi, D. Papadias, K. Tan. An Adaptive Peer-to-Peer Network for Distributed Caching of OLAP Results. ACM-SIGMOD 2002

[RD01] A. Rowstron and P. Druschel. Pastry: Scalable, Decentralized Object Location and Routing for Large-scale Peer-to-peer Conference on Distributed Systems Platforms (Middleware 2001). Heidelberg, Germany, November 2001.

[SMK01] I. Stoica, R. Morris, M. Kaashoek, and H. Balakrishnan. Chord. A Scalable Peer-to-peer Lookup Service for Internet Applications, ACM SIGCOMM 2001, August 27-31, 2001, San Diego, California, USA, pp149~160

[UB]http://xenal.ddns.com.nus.edu.sg/p2p/

[UF] http:// freenet.sourceforge.com/

[UG] http://gnutella.wego.com/

[UL] http://lockss.stanford.edu/

[US] http://setiathome.ssl.berkely.edu/

[YG01] B. Yang and H. Garcia-Molina. Comparing Hybrid Peer-to-Peer Systems, VLDB 2001, September 11-14, 2001, Roma, Italy, pp561~570

[YG02] B. Yang and H. Garcia-Molina. Efficient search in peer-to-peer networks, ICDCS 2002