Due Time Setting for Peer-to-Peer Retrieval of Teaching Material in Cloud Computing Environments

Wen-Chung Shih, Shian-Shyong Tseng*
Department of Applied Informatics and Multimedia
Asia University
Taichung, 41354, Taiwan
{wjshih, sstseng}@asia.edu.tw

Chao-Tung Yang
Department of Computer Science
Tunghai University
Taichung, 40704, Taiwan
ctyang@thu.edu.tw

Abstract— **Cloud computing technologies can be used to store a tremendous amount of learning resources, including educational videos. However, it is difficult for users to retrieve relevant teaching material in cloud environments by submitting a few keywords. We propose a search mechanism based on social networks. Furthermore, the issues of availability and trustworthiness have to be considered in teaching material retrieval process, in order to improve the retrieval performance. Current P2P file sharing applications focus on exact matching and ignore the dynamic nature inherent in P2P networks, which might degrade the retrieval performance in terms of precision and response time. Our idea is to appropriately set the due time to improve response time of P2P teaching material retrieval without sacrificing much precision. In this study, we formulate the due time setting problem as a bi-objective optimization problem and approximately solve it by an interaction-based heuristic method. The proposed approach consists of two phases: construction phase and sharing phase. In construction phase, peer information is acquired and managed in a decentralized manner. In sharing phase, the due time is interactively determined and the results are retrieved. A prototype of cloud platform has been constructed and experiments have been conducted to evaluate the performance of the approach. The experimental results show that the interactive algorithm performs well in P2P networks with low trust and availability. Also, a survey of satisfaction shows that the proposed method is more user-friendly.**

Keywords- Teaching material search; Due time setting; Cloud computing; Availability; Trustworthiness

I. INTRODUCTION

With the flourishing development of information technologies, e-learning has become a promising learning paradigm. A number of e-learning researches aim to facilitate adaptive learning, which provides a customized environment according to students' requirements. To conduct adaptive instruction, teachers need to prepare customized TM (teaching material) for students with various learning styles, which is a heavy burden for teachers. TM sharing has been proposed to avoid redundant efforts on TM authoring. However, two limitations of current TM sharing platforms hinder the development of TM sharing. First, centralized management in these platforms can assure the quality of TM, but it can also discourage the passion for authoring. For example, not all submitted TMs can be published, resulting from collection policies and censorship. Second, the problems inherent in the client-server model, such as SPOF (single point of failure), service bottleneck, etc., can impede the expansion of the community.

Cloud computing technologies can be used to store a tremendous amount of learning resources, including educational videos. However, it is difficult for a user to retrieve relevant teaching material in cloud environments by submitting a few keywords. We propose a search mechanism based on social networks. As shown in Figure 1, each peer in the social network represents a participating teacher, and an overlay is formed to represent the neighborhood relationship of participating peers. Also, a peer can join/leave the P2P (peer-to-peer) network at any time, which characterizes the dynamic nature of P2P networks. To conduct TM sharing, a peer installs the tailor-made software, which supports Blog-like operations, such as publication, comment/reply, content organization, etc. In addition, the functions of TM search and download are incorporated.

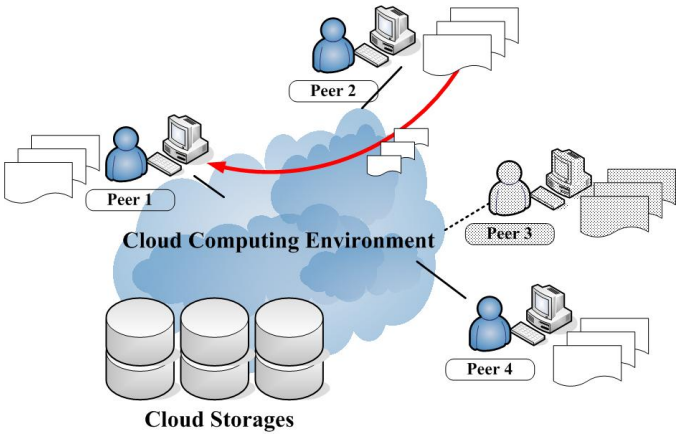

Figure 1. A TM sharing cloud environment based on social networks

Lacking a centralized index, search becomes a challenging issue in social networks. A number of researchers have been devoted to the study of P2P search. However, existing P2P search methods, which are mainly designed for file sharing applications, are not suitable for TM retrieval. The primary

978-1-4244-9224-4/11/$26.00 ©2011 IEEE

reason lies in fundamental differences between TM retrieval and file sharing in P2P networks.

Furthermore, the attitude toward response time is different for TM retrieval and P2P file search. Reducing response time is an important criterion for P2P search. However, this criterion is neglected by current P2P file sharing applications. For example, a typical P2P search process can be described as follows. The requesting peer sends a query message to some peers. Then, this peer waits for a period of time, called Due Time, to collect the response messages. In conventional P2P applications, due time is usually set a large value because of the all-or-nothing nature of exact matching. That is to say, users are willing and have to wait a long time for the desired file. For TM retrieval, however, we want to find a satisfiable TM, not a specific one. Therefore, we can pursue the goal of reducing response time.

Due time setting is one of important issues for TM retrieval in P2P networks. The primary reason is that a criterion is required to make decisions when a submitted query is not responded. Particularly, if the query is not replied in the expected time, which can be estimated by log mining, the user will be confused. In this work, the due time setting problem (DTS) is formulated as follows. Given a query, a positive integer k and a similarity function, determine the due time for each forwarded peer to retrieve k relevant teaching materials from a P2P network, where the peers might be unavailable or untrustworthy. The Goal is to maximize similarity and minimize due time. There are two main difficulties in the DTS problem. First, it is not totally a technical problem. TM retrieval involves subjective human factors. For example, the system can calculate the estimated due time according to availability information, and provide it to the user as a suggestion. It is fine when the requested peer answered in time. However, what should the user act if the requested peer does not respond in time? When unexpected situations happen, one flexible solution is to resort to users' wills. Second, too little information about peers for decision making is available. A mechanism of peer information acquisition is needed for human decision makers.

Our idea is to interactively set due time when an exception happens. Objectively, availability information can be for reference. Subjectively, it depends on users to make decisions in due time setting. For example, some user might want to wait longer for a quality peer, even with low availability. To realize this idea, a two-phased framework is proposed, which consists of a construction phase and a sharing phase. First, due time is set to a default value estimated by the system. When the query is due, the system asks the user whether to extend the due time or not. Other information is also provided for decision making.

The evaluation of the performance for the proposed approach consists of quantitative experiments and qualitative surveys. In the quantitative experiments, the interactive algorithm is compared with the two greedy algorithms in terms of similarity and waiting time. In the qualitative surveys, a satisfaction survey is conducted to understand the degree of user satisfaction for the design of interactive due time setting. The results imply that the interactive method is more flexible than the two greedy algorithms, and can be accepted by most of the users.

The contributions can be concluded as follows. First, we propose a Blog-like P2P TM sharing platform. Second, the Due Time Setting problem is formulated and solved. Particularly, a decentralized peer information management scheme is designed. Most importantly, an interactive due time setting algorithm is proposed.

The rest of this paper is organized as follows. In Section II, we review the preliminaries and previous work on related research. Then, the problem is formulated in Section III. Next, the approach is presented in Section IV. Implementation and experimental results are discussed in Section V. Finally, the concluding remarks are given in Section VI.

II. PRELIMINARIES AND RELATED WORK

In this section, the background knowledge of cloud computing software is reviewed. Then, TM retrieval and P2P search are compared and related researches are surveyed.

A. Hadoop and HDFS

Hadoop is one of the most salient pieces of the data mining renaissance which offers the ability to tackle large data sets in ways that weren't previously possible due to time and cost constraints. It is a part of the apache software foundation and its being built by the community of contributor in all over the world. The Hadoop project promotes the development of open source software and supplies a framework for the development of highly scalable distributed computing applications [1].

Hadoop is the top-leveled project in Apache Software Foundation and it supports the development of open source software [2]. Hadoop provides a framework for developing highly scalable distributed applications. The developer just focuses on applying logic instead of processing detail of data sets. The HDFS (Hadoop Distributed File System) file system stores large files across multiple machines. It achieves reliability by replicating the data across multiple hosts, and hence does not require RAID storage on hosts. The HDFS file system is built from a cluster of data nodes, each of which serves up blocks of data over the network using a block protocol. They also serve the data over HTTP, allowing access to all content from a web browser or other client. Data nodes can connect to each other to rebalance data, to move copies around, and to keep the replication of data high. A file system requires one unique server, the name node. This is a single point of failure for an HDFS installation. If the name node goes down, the file system will be off-lined. When it comes back up, the name node must replay all outstanding operations. This replay process can take over half an hour for a big cluster [3].

B. TM Search and P2P Search

General information retrieval methods are mainly designed for web pages. Nevertheless, documents in specific domains may need tailor-made methods to improve their retrieval performance. For example, FAQ search [4-6] and patent

retrieval [7-9] are widely investigated to find more efficient methods.

TMs are also a special kind of documents. They are distinct in two aspects. First, they are for educational purposes. Second, they have standardized format, such as Learning Objects, Content Packages. To share and reuse teaching materials, several standards have been proposed recently. Among these, SCORM (http://www.adlnet.org/) is the most popular standard for learning contents. It was proposed by the U.S. Department of Defense's Advanced Distributed Learning (ADL) organization in 1997. This standard consists of several specifications developed by IEEE LTSC (Learning Technology Standards Committee, http://ltsc.ieee.org/wg12/), IMS (Instructional Management System, http://www.imsproject.org/), AICC (Aviation Industry CBT Committee, http://www.aicc.org/), etc. SCORM Metadata refers to the IEEE's LOM (Learning Object Metadata, http://ltsc.ieee.org/wg12), and describes the attributes of teaching materials. IEEE LOM v1.0 includes nine categories: General, LifeCycle, Meta-Metadata, technical, educational, rights, relation, annotation, and classification.

Search in P2P networks has been a flourishing research topic in recent years. A number of solutions to P2P search have been proposed [10-12]. Zhu et al. [10] indicated that one of the difficulties in P2P search is the lack of global statistical information, thus impeding the straightforward application of the well-known vector space model approach to P2P networks. Content search is addressed in [13, 14]. In [10-12], information retrieval methodologies were conducted in P2P networks. In [15], they studied social network. Table I compares related researches and systems.

TABLE I. CLASSIFICATION OF P2P SEARCH APPROACHES

Approaches	Description	Disadvantages	Examples
Centralized Indexing	A global index is maintained for efficient search.	SPOF	Napster
Flooding	The requester broadcasts the query across the whole network.	Poor network utilization	Gnutella
DHT-based (Distributed Hash Table)	A hash table is maintained in a distributed manner to decide the location of a file.	Only supporting exact matching	Chord [16]

Most of these researches ignored information retrieval techniques. In addition, the issues of availability and trustworthiness have not been considered. Therefore, TM search in P2P networks requires a more suitable solution. Traditionally, TM retrieval approaches can be categorized into: keyword-based search and metadata-based search. In this paper, we retrieve TMs based on both the content keywords and metadata.

III. PROBLEM FORMULATION

In essence, information retrieval is a kind of human activities, which can be more efficient with the assistance of computer systems. The goal of traditional information retrieval problems is to improve such measures as precision, recall, etc. However, when considering TM retrieval in P2P networks, we should include more potential goals, such as response time. Therefore, we formulate the due time setting problem as a bi-objective optimization problem. Due to the dynamic nature inherent in P2P environments, we do not intend to formulate this problem in a strict mathematical form, which is used to be solved in an evolutional computing approach. Instead, we present a formulation which is suitable for interaction-based heuristic solutions. As explained in Section 1, the due time setting problem is to determine the due time for each peer to whom messages are forwarded. In this section, we introduce related definitions and then formulate this problem.

Traditional, similarity is measured by the Vector Space Model (VSM) in information retrieval domain [17, 18]. In the VSM-based model, a document is represented as a vector. In general, a limited vocabulary of keywords is adopted to denote important words in documents. Each element of the vector corresponds to a keyword of the vocabulary. Therefore, the length of the vector for a document is equal to the size of the vocabulary. The value of each element is a weight denoting the importance of the keyword to the document. There are a number of methods to determine the weights. Among them, TF-IDF [19] is the most well-known method to assign weights. The TD-IDF method is based on two findings. First, words frequently used in a document, except stop words, are important keywords with respect to this document. Second, words frequently appearing in many documents are not important for the purpose of differentiation.

As standardization of teaching materials becomes a trend, we model TMs by the SCORM standard, where a Content Package (CP) is defined as a package of learning materials, and a Peer Repository (PR) is a set of Content Packages stored and published by a peer. The content package is represented by a vector of keyword weights. To enable content-based retrieval, the Vector Space Model is applied to represent the text content and to calculate the keyword weights. Also, the Educational category of the LOM metadata is included in this model of CP. Therefore, a TM representation consists of a vector of weights and a category of metadata.

A Query is used by a user to specify the TMs s/he wants. Users can express their queries in two forms: keyword-based and metadata-based. A keyword-based query is a vector of keyword weights, which mean the concepts about the desired contents. A metadata-based query is a list of (Attribute, Value) pairs, which describe the properties of TMs.

In order to determine the degree of relevance of a query and a teaching material, the similarity function has to be defined. Conventional similarity functions, such as the cosine function,

are not suitable for SCORM-compliant teaching materials which are characterized by textual content, metadata and structural information. Here, a similarity measure Sim between a query Q and a teaching material TM is proposed by combining a keyword-based similarity and a metadata-based similarity. The keyword similarity Sim_K adopts a cosine function to measure the text similarity between a query and a TM. The metadata similarity Sim_M is defined to be the number of matched attributes divided by the number of all attributes. Therefore, the range of these two similarity terms, Sim_K and Sim_M, are both in [0, 1]. The similarity measure Sim is defined in (1).

$$\mathrm{Sim}(Q,TM) = \alpha \times \mathrm{Sim}_K(Q,TM) + (1-\alpha) \times \mathrm{Sim}_M(Q,TM) \quad (1)$$

where the factor α, $0 < \alpha < 1$, is used to control the relative weighting of keyword similarity and metadata similarity. The setting of the α value is discussed in Section 5.

The P2P environment proposed in this work is an unstructured P2P network, where there is no centralized index structure. We model this P2P network as a graph. A node represents a peer who publishes TMs in local storage. An edge means a friendship relation between two peers. The network formed by the friendship relation is also called a P2P overlay network. To characterize the dynamic nature of the P2P network, where a peer can join/leave the network at any time, two important concepts are defined. Availability means the probability that a peer is on-line at some time instance. Trust is the probability that the information published by a peer is correct.

When a peer submits a query, it also assigns a due time, when the requesting peer will begin to merge results. After due time, responses from requested peers will not be accepted. The Due Time Setting (DTS) problem is described as follows. Given a query Q, a positive integer k and a similarity function Sim, determine the due time T_{Due} for retrieving k relevant teaching materials from a P2P network, where each peer has a repository, and the peers might be unavailable or untrustworthy. The goal is to optimize the following two objective functions:

$$\text{Max} \quad \frac{1}{k}\sum_{i=1}^{k} Sim(Q, TM_i) \quad (2)$$

$$\text{Min} \quad T_{\text{Due}} \quad (3)$$

where TM_i is the i-th TM retrieved from the P2P network.

IV. APPROACH

As mentioned in Section 1, the main difficulty of the DTS problem lies in that it is not totally a technical problem. TM retrieval involves subjective human factors, and flexible setting is desirable to respond to unexpected situations in P2P networks. In addition, too little information about peers for decision making is available. Therefore, our idea is that users can interactively adjust the due time. The architecture and algorithms based on this idea are presented in this section.

A. Peer Node Architecture

To realize this idea, a two-phased architecture for each peer node is proposed. In Construction Phase, an overlay network is formed, and the peer information is collected and managed. In Sharing Phase, the peer node can publish and retrieve TMs. The four modules are described as follows:

TM Publication Module. This module enables this peer to publish TMs in a Blog-like manner. The published TMs are posted on the "Blog", and are stored in the local repository. Other peers can browse and download the published TMs. In addition, common Blog operations are supported by this module, such as comments.

TM Retrieval Module. This module enables the peer to retrieve desired TMs from the P2P network by submitting a query.

Peer Information Management Module. The purpose of this module is to support decision making. This module manages the peer's information, including private information and public information. The former is used by the peer, and the latter is for public access.

Overlay Construction Module. This module enables the peer to develop a friendship overlay in the P2P network. A number of methods have been proposed to build P2P overlay, among which the method proposed by [11] is an effective method. In this work, we use existing methods to implement the overlay construction module.

Lacking global information, each peer maintains three data structures in this architecture. 1) A Friend table. This table is built by the overlay construction module. Each row represents a friend peer. The table size is limited by the capacity of system memories. The fields include Trust, Availability, Collection Summary (*C_ Summary*) and Collection Size (*C_Size*).

- Trust. A peer evaluates this value for each friend. The trust value is an accumulative score about the trustworthiness of the friend. The update of Trust is described in the next section.
- Availability. A peer maintains this value for each friend. The peer acquires the availability value which is published by the friend. Therefore, the accuracy of the value depends on the friend's trustworthiness. A trustworthy peer will probably provide correct availability.
- Collection Summary. Each peer publishes the summary information of its repository, in order that others can access this summary information and estimate the content of the repository. This summary information is also represented by a VSM-based vector, as described in Section 3. For example, assume that the vocabulary includes three keywords: K_1, K_2 and K_3. A peer can publish its collection summary by calculating the three weights, w_1, w_2 and w_3, from TMs in its repository. Then, the summary vector is obtained as $< w_1, w_2, w_3 >$.
- Collection Size. This value is also provided by each peer for others to access. For example, if a peer has

20 TMs in its repository, this peer can publish its Collection Size as 20.

An example is shown in Table II. 1) This peer maintains three friends: peer 2, peer 3 and peer 4. 2) Self-information array. This array contains such information as Availability, C_Summary and C_Size, which is for public access. 3) A Local index for its TM collection. With this index, the search in local repository can be sped up. The index is built by the scheme in [20].

TABLE II. AN EXAMPLE OF A FRIEND TABLE

	Trust	Availability	C_Summary	C_Size
Peer 2	0.9	0.2	<0.7, 0.6, 0.4>	20
Peer 3	0.6	0.8	<0.5, 0.8, 0.6>	35
Peer 4	0.5	0.6	<0.3, 0.7, 0.9>	16

B. Peer Information Management

Peer information management is an important task in Construction Phase. To facilitate decision making for TM retrieval in the P2P network, each peer maintains two categories of information:

- Public information. This kind of information is about the peer itself, and is generated by itself for other peers to access. Availability, collection summary and collection size can be categorized to this class.
- Private information. Each peer privately collects this type of information from other peers, such as trust information. This information can be obtained by evaluating previous searching transactions.

Trust information represents the degree to which the peer has trust in the information provided by other peers. Trust information can be used to update friend tables and evaluate other peers' information. Policies for initializing trust information depend on peers. The optimistic policy can have 1 as the initial value. The pessimistic policy sets the initial value to 0. With a neutral policy, the initial value is 0.5. Updating of trust information occurs after each searching transaction. If the retrieved TM from peer *j* has a higher similarity value than the *C_Summary* claimed by peer *j*, the Trust for peer *j* will be increased. Otherwise, the trust score will be decreased. A normalization function, Normal, will be applied to the calculated score to generate a value ranging from 0 to 1. The definition of Normal is as follows. It returns:

- 1, if its input parameter is larger than 1;
- 0, if its input parameter is smaller than 0;
- its input parameter, otherwise.

We denote the original trust information for peer *j* by $Trust_j$, and the updated trust by $Trust_j$'. Also, Sim is the similarity function defined in (1). Then the update formula is listed as follows.

$$Trust_j' = Normal\,(Trust_j + Sim(\mathrm{Q}, \mathrm{V_R}) - Sim(\mathrm{Q}, \mathrm{V_C})) \quad (4)$$

where

- Q is the query;
- V_R is the content vector for the retrieved TM;
- V_C is the collection summary information published by peer *j*.

Availability information, generated by a peer itself, means the probability that a peer is on-line. Methods for generating availability information depend on peers. Simple methods can estimate online probability by statistics on the past 24 hours. Advanced mechanism, such as Markov Process, can also be used to generate predictive online probability. Update frequencies also depend on peers.

Collection Size is the number of TMs in the peer's collection. Collection Summary is the summary information of the collection. The degree of detail depends on peers' choice. An average vector, level-wise vectors and Categorized vectors provide different details.

C. TM Retrieval

In contrast to traditional P2P search methods, such as flooding, we adopt a query forwarding mechanism based on a built overlay to improve the search performance. The algorithm for TM retrieval consists of the following main steps: local search, query forwarding and result merging. Query forwarding includes peer selection and due time setting. This section focuses on the former, and the latter is presented in the next section.

The query forwarding is based on the P2P overlay. That is, a peer forwards the query to its selected friends. The purpose is to select peers which possibly have more similar TM. The criterion for peer selection is as follows.

$$Trust_j \times Sim(\mathrm{Q}, \mathrm{V_j}) \quad (5)$$

where

- $Trust_j$ is trust information of peer *j*;
- V_j is the vector of collection summary for peer *j*.

In Result Merging, the requesting peer collects the responses from its friends. When the request is satisfied, the requesting peer can choose to finish the merge and omit the friends who are still in processing. Condition for stopping searching beforehand can be stated as follows.

- The number of retrieved TMs >= k, and
- The estimated similarity of the remaining peer < the smallest similarity of retrieved TMs

When a peer receives a searching request, it executes this algorithm, Peer_Retrieval. Step 2.2 calls the subroutine Sub_DTS to interactively set the due time, which is detailed in the next section.

Algorithm: Peer_Retrieval (ALG_PR)

Symbols Definition:

Q: the query submitted by a user

Num_TM: the number of TMs to be retrieved

TM_set_size: the size of the returned set of TMs

TM_set_summary: summary of the returned set of TMs

Input: Q, Num_TM

Output: TM_set_size, TM_set_summary

Step 1: Local searching

Step 2: Query forwarding

Step 2.1: Select peers

Step 2.2: call Sub_DTS // for due time setting

Step 2.3: Forward the query

Step 3: Result merging

Step 4: Trust information updating

Step 5: Return

D. The Heuristic Method for Due Time Setting

This section describe the subroutine called by Step 2.2 of ALG_PR for due time setting in detail. This method is based on an IRT (Item Response Theory)-like function, which characterizes the relationship between due time and availability. Let $Prob_j(T_{due})$ denote the probability that a requester can access a peer j given the setting of due time, T_{due}.

$$Prob_j(T_{due}) = \frac{1}{1+e^{(T_{due}-T_{avail})}} \quad (6)$$

where T_{due} is the time instance before which the requester will wait; T_{avail} is the estimated time instance before which the peer j will not be available. For example, assume that T_{avail} is 8:30 p.m. If Tdue is set to 8:30 p.m., the probability of successful access to peer j is 0.5. When $T_{due} > T_{avail}$, the probability will be larger than 0.5. When $T_{due} < T_{avail}$, the probability will be smaller than 0.5.

For initialization, system default values are calculated according to the formula (6). During the search process, users can interactively extend the default due times. If a quality peer is not on-line, the system will ask the user whether to extend the due time or not. If an on-line peer exceeds the due time, the system will also ask the user whether to extend the due time or not. For example, "Peer 2 is not on line. It has quality TMs you want, and might get on line soon. Would you extend the due time?" The procedure is listed as follows.

Subroutine: Interactive_Due_Time_Setting (Sub_DTS)

Symbols Definition:

T_{due}: the time instance before which the requester will wait

T_{avail}: the estimated time instance before which the peer j will not be available

$Prob_j(T_{due})$: the probability that a requester can access a peer j given T_{due}

p: the desired probability that a requester can access a peer j

Input: p, T_{avail}

Step 1: Set Tdue according to the formula of $Prob_j(T_{due})$ in (6), given p.

Step 2: Wait until time is due.

Step 3: For all peers who have not responded

Step 3.1: Show the unresponsive peer's information, and ask the request whether to wait or not.

Step 3.2: Get the requester's answer.

Step 4: If the requester will not wait, return.

Step 5: Go to Step 1.

V. EXPERIMENTAL RESULTS

In this section, the implementation and evaluation design are described. Then, experimental results are presented and discussed.

A. Test Collections and P2P Configurations

We have collected two TM sets for the experiments, SLN and LFS. SLN is a collection of TM transferred from the Six Learning Nets (http://learning.edu.tw/sixnet/) built by the Ministry of Education, Taiwan. LFS is a collection of TM transferred from the Learning Fueling Station (http://content1.edu.tw/) built also by the Ministry of Education, Taiwan. Characteristics of the two test collections are presented in Table III.

TABLE III. CHARACTERISTICS OF THE THREE TEST COLLECTIONS

Collection / Characteristics	SLN	LFS
No. of TMs	1200	1200
Average Length of a TM (word)	451.3	1050.2
Subject	Mathematics	Mathematics
Metadata	Yes	Yes

We have simulated four P2P environments with different trust and availability. In P2P configuration with low trust, the probability that a peer honestly publishes its information is 0.25, while the probability is 0.75 in a high-trust P2P configuration. Similarly, In P2P configuration with low availability, the probability that a peer is on-line is 0.25, while the probability is 0.75 in a high- availability P2P configuration. Characteristics of the four configurations are presented in Table IV.

TABLE IV. CHARACTERISTICS OF THE FOUR P2P CONFIGURATIONS

P2P Configuration	Trust	Availability
1	High	High
2	Low	High
3	High	Low
4	Low	Low

(low = 0.25, high = 0.75)

We have implemented a video sharing platform to conduct the experiments, as shown in Figure 2 and 3. This prototype is developed based on open-source software, Hadoop (http://hadoop.apache.org/). We have also setup a small-scale P2P community, which consists of twelve elementary-school teachers.

Figure 2. A video sharing platform based on Hadoop cloud computing software

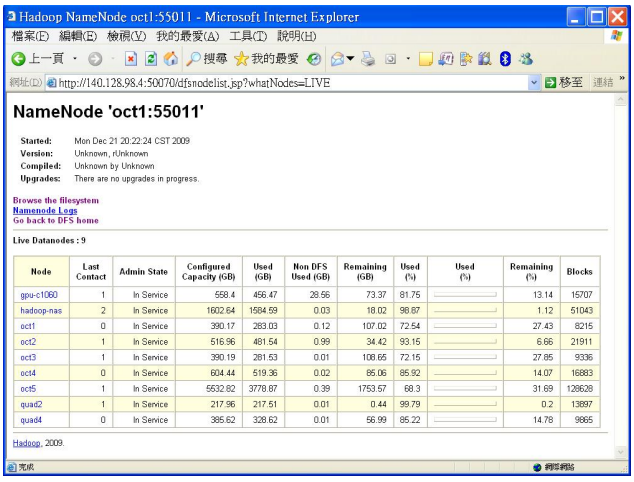

Figure 3. The snapshot of the monitoring tool on the cloud

B. *Quantitative analysis*

We have designed another two greedy DTS methods based on precision and response time respectively, for the purpose of comparison with the interaction-based method. The precision-based method always waits for all friends to respond the query, while the time-based method waits for a pre-defined time period. The measure of precision is the same as that defined in conventional information retrieval literatures.

$$Precision = N\_Relevant / N\_Retrieved \quad (7)$$

where

- $N\_Relevant$ is the number of relevant TMs in the retrieved TMs;
- $N\_Retrieved$ is the number of retrieved TMs.

The two figures in Figure 4 show the results with respect to SLN and LFS respectively. First, we observe that the precision-based method performs well in all configurations. This is because it broadcasts to all peers, and waits a long time for the results. This will incur a large cost. Second, the time-based method degrades significantly when the P2P configuration becomes low-trustworthy or low-available. The main reason may be that it statically set a small due time, and some relevant results are too late to be received. The interactive method is almost as good as the performance-based method in terms of precision.

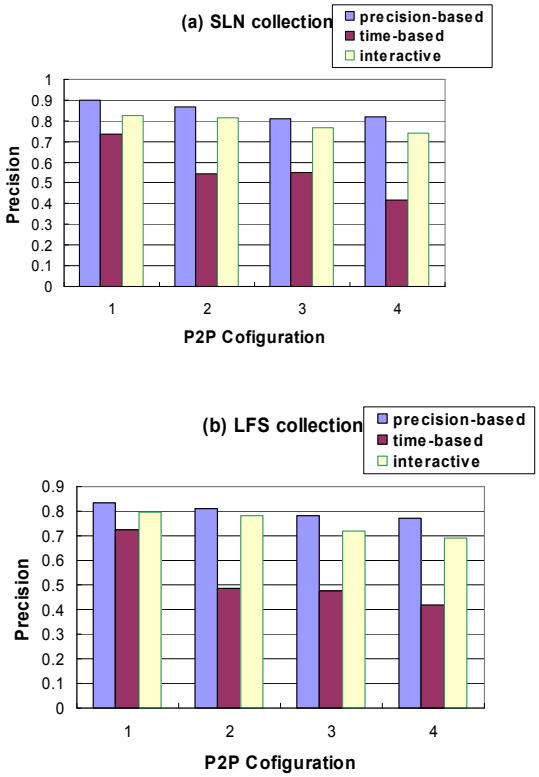

Figure 4. Comparison of Precision for the three DTS methods: (a) SLN; (b) LFS.

The two figures in Figure 5 show the results in terms of response time. First, we observe that the time-based method performs well in all configurations. This is because it sets a fixed small due time. However, this will incur poor precision as mentioned above. Second, the precision-based method

degrades significantly when the P2P configuration becomes low-trustworthy or low-available. The main reason may be that it always set a large due time. When the environment becomes dynamic, the response time will be long. The interactive method is almost as good as the performance-based method in terms of response time.

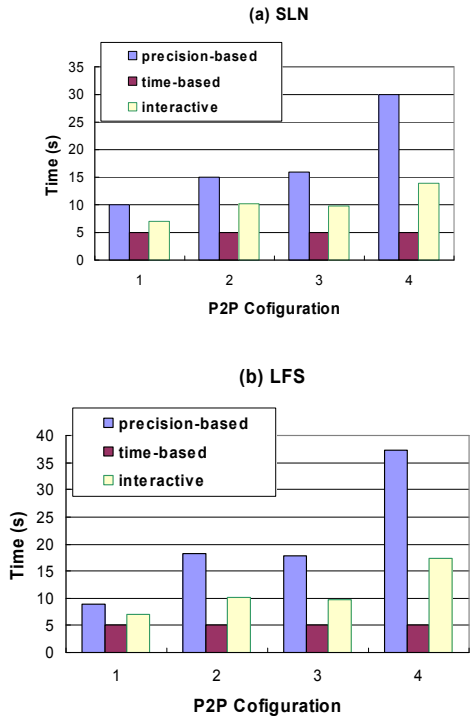

Figure 5. Comparison of response time for the three DTS methods: (a) SLN; (b) LFS.

The two experiments show that the interactive algorithm is both effective in terms of precision and response time. Although it is not the optimal method, the interactive method can adapt to low-availability and low-trust P2P networks.

C. *Qualitative analysis*

One month after the users began to use this application, a survey was conducted with respect to the twelve teachers. Each teacher was asked four categories of questions to obtain their comments on the prototype system. Each category has five questions, and a five-point Likert scale with anchors ranging from strongly disagree (1) to strongly agree (5) is used for this survey. The mean value and standard deviation (SD) is calculated for each category.

For Category 1 questions, the deviation of user satisfaction is slightly larger than other categories. The reason may be that the participants are not all familiar with the concept of Blog. Some participants comment that they are not used to publish their articles to others. However, some participants appreciate this idea and like to frequently update and publish their TMs.

The results of Category 2 and 3 show that participants are willing to publish their private information for efficient retrieval. However, the published information is not highly satisfiable. This implies that we can reconsider to include more useful information as public information to improve the user satisfaction, such as peers' professional information.

In summary, the interactive due time setting is user-friendly and satisfactory, according to the results of Category 4.

TABLE V. THE RESULT OF THE SURVEY

Category No.	Questions	Mean	SD
1	Satisfaction of the blog-like TM sharing platform	3.46	1.23
2	Satisfaction of peer information provided by others	3.69	0.91
3	Willingness to provide self information	4.62	1.18
4	Satisfaction of interactive due time setting	4.33	0.55

D. *Discussion*

As indicated in (1), this study adopts the similarity measure which combines keyword similarity and metadata similarity through a controlling factor, α. In this section, we investigate how the α value is determined to improve the precision of TM retrieval. Twelve queries are submitted to SLN and LFS collections, respectively. Also, the α value is set to be 0, 0.25, 0.5, 0.75, 1, respectively. For each different setting, the experiment is repeated five times, and the average precision is obtained. To avoid the dynamic effect of P2P networks, all participating peer are always on-line.

Figure 6 shows the precision value for different α values, which range from 0 to 1. The observations can be summarized as follows. First, we obtain the best precision when α = 0.75, among the five α values. This implies that the combination of keyword similarity and metadata similarity is useful. Second, the precision on α = 0 (metadata similarity only) is slightly smaller than that on α = 1 (keyword similarity only). This probably results from the characteristics of the two TM collections. The metadata do not include detailed subject information. Therefore, using merely metadata can not exactly retrieve the desired TM. Based on the results, we set α = 0.75 for experiments in this paper.

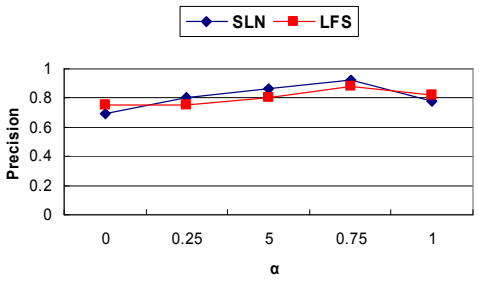

Figure 6. The effect of the α value on precision

VI. Conclusions and Future Work

In this study, Blog-like P2P environments are proposed and shown as promising platforms for TM sharing. To reduce response time without sacrificing retrieval precision, we formulate the due time setting as a bi-objective optimization problem, and propose an interaction-based heuristic algorithm to approximately solve it. In addition, a decentralized architecture is designed to facilitate the acquisition and management of peer information. Due to the dynamic nature inherent in P2P systems, issues of availability and trustworthiness are considered in the proposed algorithms.

Experimental results show that the dynamic nature of a P2P system will degrade the performance of retrieval algorithms if they do not take the issues of availability and trustworthiness into account. Advantages of the proposed algorithm can be summarized as follows. First, the due time setting algorithm improves response time without sacrificing much precision. Based on the effective management of trust, availability and collection information, the dynamic status of peers can be estimated. Second, the due time setting in an interactive manner redeems the possible failure in statistical estimation, and provides a flexible way for human users to make decisions of due time extension.

In the near future, we plan to extend this work for investigating interaction between human and systems in TM retrieval. Furthermore, we will try to integrate the Blog-like sharing platform with Wiki-based TM designing, which is our previous work.

Acknowledgment

This research was partially supported by National Science Council of Republic of China under the number of NSC97-2511-S-468-004-MY3, NSC98-2511-S-468-004-MY3, NSC98-2511-S-468-002 and NSC99-2511-S-468-003.

References

[1] J. Venner, *Pro Hadoop*, 1st Edn ed.: Apress, 2009.

[2] "Apache Hadoop Project."

[3] R. Grossman, "Compute and storage clouds using wide area high performance networks," *Future Generation Computer Systems,* vol. 25, pp. pp. 179-183, 2009.

[4] C.-H. Wu, J.-F. Yeh, and Y.-S. Lai, "Semantic Segment Extraction and Matching for Internet FAQ Retrieval," *Transactions on Knowledge and Data Engineering,* vol. 18, pp. 930-940, 2006.

[5] H. Kim, H. Lee, and J. Seo, "A reliable FAQ retrieval system using a query log classification technique based on latent semantic analysis," *Information Processing and Management,* vol. 43, pp. 420-430, 2007.

[6] H. Kim and J. Seo, "High-performance FAQ retrieval using an automatic clustering method of query logs," *Information Processing and Management,* vol. 42, pp. 650-661, 2006.

[7] S. Fujita, "Technology survey and invalidity search: A comparative study of different tasks for Japanese patent document retrieval," *Information Processing and Management,* vol. 43, pp. 1154-1172, 2007.

[8] I.-S. Kang, S.-H. Na, J. Kim, and J.-H. Lee, "Cluster-based patent retrieval," *Information Processing and Management,* vol. 43, pp. 1173-1182, 2007.

[9] Y. Li and J. Shawe-Taylor, "Advanced learning algorithms for cross-language patent retrieval and classification," *Information Processing and Management,* vol. 43, pp. 1183-1199, 2007.

[10] X. Zhu, H. Cao, and Y. Yu, "SDQE: towards automatic semantic query optimization in P2P systems," *Information Processing and Management,* vol. 42, pp. 222-236, 2006.

[11] J. X. Parreira, S. Michel, and G. Weikum, "p2pDating: Real life inspired semantic overlay networks for Web search," *Information Processing and Management,* vol. 43, pp. 643-664, 2007.

[12] H. Nottelmann and G. Fischer, "Search and browse services for heterogeneous collections with the peer-to-peer network Pepper," *Information Processing and Management,* vol. 43, pp. 624-642, 2007.

[13] H. T. Shen, Y. Shu, and B. Yu, "Efficient semantic-based content search in P2P network," *Transactions on Knowledge and Data Engineering,* vol. 16, pp. 813-826, 2004.

[14] D. Zeinalipour-Yazti, V. Kalogeraki, and D. Gunopulos, "pFusion: A P2P Architecture for Internet-Scale Content-Based Search and Retrieval," *Transactions on Parallel and Distributed Systems,* vol. 18, pp. 804-817, 2007.

[15] S. J. H. Yang and I. Y. L. Chen, "A social network-based system for supporting interactive collaboration in knowledge sharing over peer-to-peer network," *International Journal of Human - Computer Studies,* vol. 66, pp. 36-50, 2008.

[16] S. Ion, M. Robert, L.-N. David, R. K. David, M. F. Kaashoek, D. Frank, and B. Hari, "Chord: a scalable peer-to-peer lookup protocol for internet applications." vol. 11: IEEE Press, 2003, pp. 17-32.

[17] G. Salton and M. J. McGill, *Introducion to Modern Information Retrieval*. New York: McGraw & Hill, 1983.

[18] R. Baeza-Yates and B. Ribeiro-Neto, *Modern information retrieval*. New York: ACM Press, 1999.

[19] G. Salton, A. Wang, and C. Yang, "A Vector Space Model for Information Retrieval," *Journal of the American Society for Information Science,* vol. 18, pp. 613-620, 1975.

[20] W.-C. Shih, S.-S. Tseng, and C.-T. Yang, "Using Taxonomic Indexing Trees to Efficiently Retrieve SCORM-compliant Documents in e-Learning Grids," *accepted by Journal of Educational Technology & Society,* 2008.