Asian Journal of Information Technology 13 (3): 138-149, 2014
ISSN: 1682-3915

Dynamic Caching for Semantic Oriented Adaptive Search in Unstructured P2P Networks

G. Ramachandran and K. Selvakumar
Department of Computer Science and Engineering,
Annamalai University, Annamalainagar-608 002, Tamil Nadu, India

Abstract: Most of the existing unstructured Peer to Peer (P2P) supports the filename or keyword based limited search techniques. This study proposes a novel Semantic Oriented Adaptive Search (SOAS) strategy based on semantic content. It is a multi-layer architecture model which utilize a dynamic caching technique to achieve effective search on unstructured P2P network. The proposed scheme constructs as a two-tier P2P network with ultra peers of high connectivity based on a power law model. The novelty of the proposed scheme is that the query is processed through multi-tier summary indexing framework. The proposed approach extensively used Vector Space Model (VSM) and Latent Semantic Index (LSI) to derive local indices from summarized semantic vectors. Query searching concedes through a round of searches using derived indices from semantic data objects. If one search fails, the next round search is invoked sequentially in the order of local index, cache index, response index, global index and adaptive search among ultra peers. The proposed SOAS produces a high success rate and generates a minimal amount of network traffic for an effective content search. Dynamic Time to Live (TTL) based cache consistency is proposed where each ultra peer dynamically caches the responses of previously requested queries based on the query popularity rate and TTL. An implementation and large scale simulation are performed to evaluate the proposed approach. The experimental result proves that the proposed system performs better than the existing approach in terms of accuracy, response time, success rate and cache hit.

Key words: Peer to peer search, semantic search, vector space model, latent semantic index, dynamic caching

INTRODUCTION

P2P computing has emerged as a powerful model to share large volumes of data between the peers directly in the network (Milojicic *et al*., 2003). Peers in unrelated administrative domains are self-organized into overlay networks in which nodes are end systems on the internet. It stores the information of a group of neighbor nodes in the P2P layer. Therefore, these nodes result in a virtual overlay on the internet axes. The P2P network is differentiated into two types based on the network topology and the degree of centralization (Lua *et al*., 2005). Depending on the degree of centralization, it can be distinguished into purely decentralized (Gnutella, Freenet), partially centralized (Kazaa, Morpheus) and Hybrid decentralized (Napster). On the other hand, depending on the network topology, it can be classified into three types such as structured, unstructured and loosely structured. P2P Search algorithms are categorized into two types as informed (Heuristic) and uninformed (Blind) based on usage of location resources. In unstructured networks, uninformed search algorithms are used to perform searches where peers are termed as a blind because it does not have the ability to search and determine neighbor peers (Dimitrios and Roussopoulos, 2003; Kalogeraki *et al*., 2002). It is hard to find the location of resources and data items in randomly placed peers in unstructured P2P networks. The existing P2P searching algorithms do not perform efficient searching because of the ineffective techniques due to decentralized architecture.

Commonly used P2P searching schemes supports name or keyword based searching that relies on matching over file names and keywords. Moreover, this search performs well only for file names but it is not efficient in the case of semantic content. Latterly, P2P network is extending with the entire federated content based search that constructs content based overlay based on small world properties that can provide a sophisticated flexible query searching to deliver a high quality of search results (Lu and Callan, 2007). Semantic content based query searching (Shen *et al*., 2004) is a novel distributed

Corresponding Author: G. Ramachandran, Department of Computer Science and Engineering, Annamalai University, Annamalainagar-608 002, Tamil Nadu, India

searching in which the two-tier hierarchical summary indexing structure is constructed in two levels; seeing that local index in each peer level and group index in ultra peer level. Similarity, metric is the dot product of the two high-dimensional point-representations of queries and documents that determine the relevance between queries and documents among the peers in the networks. In the proposed approach, specific enhancements are carried out in the existing searching schemes to achieve high search efficiency with less overhead. Instead of flooding the query throughout the network, the efficiency of the proposed method improves through forwarding the query only to the selected set of relevant peers that can provide the high quality of search results.

Contributions: The P2P overlay network is constructed in the form of two-tier where ultra peers in the upper level and client peers in the lower level through self-motivated topology transformation algorithm based on Power Law Distribution (PLD). In the process of index derivation, Vector space model summarizes the meta data of high dimensionality into semantic vectors of reduced dimensionality. Latent Semantic Indexing (LSI) derives local indices from summarized semantic vectors. Query processing is performed through well-organized summary and indexing framework in sequence of searches. Hence, there is no need to flood the query throughout the peers in a network. If one round search fails, then next round of search is automatically invoked sequentially in the order of local index, cache index, response index, global index and adaptive searching. In order to promote effective cache management with fresh content sharing, dynamic TTL based caching algorithm is proposed where a cache associated with each peer is refreshed dynamically according to the query request rate (Popularity) and TTL.

LITERATURE REVIEW

This study discusses several P2P searching algorithms based on the blind search, semantic content search and caching.

Unstructured P2P blind search: Flooding and random walk are the primary search algorithms used in unstructured P2P networks. Flooding (blind search) propagates a query among the peers in the network nevertheless limits scalability. A variation of flooding is proposed namely Adaptive Probabilistic Searching (APS), ant search, light flooding, dynamic search and differential search algorithm. In these approaches, a query is forwarded to a subset of its neighbors probabilistically to find the location of the desired data element.

APS performs searching is similar to random walkers but it differs in query forwarding functionality (Tsoumakos and Roussopoulos, 2003). In order to discover the data item, number of random walkers forward a query message based on the probabilistic information maintained in the routing table. Ant Search algorithm (Wu *et al.*, 2006) is a Controlled Flooding algorithm which solves the free riding problem in the Gnutella file sharing system. Here, queries are flooded only to peers having high potential to provide the response to requesting query. A light flooding approach evolves by Jiang *et al.* (2008) with the slight variation in message propagation from flooding to reduce the number of redundant messages in query propagation. A Dynamic Search (DS) algorithm (Tsungnan *et al.*, 2009) intelligently decides where to forward the query among the available large number of peers in the network based on the information learned from the previous experiences. Thus, searching is dynamically performed by either flooding (short term search) or random walk (long term search) using knowledge-based search mechanisms such as APS, local indices and an intelligent search. The proper setting of optimal parameters in Dynamic Searching algorithm achieves a good tradeoff between the searching performance and cost. Differential Search algorithm (Chen and Xiao, 2007) exploits the peer heterogeneity in terms of bandwidth, storage and processing capacity to improve search efficiency. Aforementioned searching approaches perform efficient searching but it relies on keyword and file name based searching that limits user search quality and requirements. In order to overcome the problem, semantic content based search techniques are proposed.

Unstructured semantic content based P2P search: Edutella (Nejdl *et al.*, 2003) is a schema based RDF P2P network that evolved in the form of ultra peer based topologies. This architecture provides an expressive and extendable description of heterogeneous services therefore it is suited for networks even with a large number of heterogeneous service providers. It achieves higher scalability than any other flooding or broadcast based networks and also performs efficient clustering functionalities among the peers in the network. Therefore, this review is carried on semantic content based information retrieval approaches. In Semantic Overlay Networks (SONs), the group of peers with similar content is self-organized under a high degree of node autonomy (Crespo and Garcia-Molina, 2005). The main challenge of SON construction is how to apply a concept classification hierarchy to each SON. In order to meet this challenge, this study provides the solution that classifies the peers according to its content in a similar group of peers. Queries are routed to the corresponding SON which has semantically relevant resources related to the query that dramatically reduces the network traffic.

To improve the robustness of P2P searching, a search is limited to a small area searching scope called as Semantic Small World (SSW) overlay network (Mei *et al*., 2011). It is constructed with peers of similar semantics and then summary indexing architecture is obtained through LSI and VSM. High dimensional semantic index leads to attract high overhead. In Gnutella network GES (Yingwu and Hu, 2006), semantically related peers are organized into the overlay structure through semantic links using a Dynamic Topology Adaptation algorithm. Therefore in search protocol, flooding performs in an overlay structure whereas the biased random search is held in semantic groups of random links. The main feature in GES System is that the vector space model improves query searching competence and excellence of search results (precision and recall) using goodness of semantic groups and effectiveness of search protocol.

A lot of existing research on Semantic Query Routing (SQR) provides relevant results only for a small collection of resources but in the case of a large collection of resources it does not provide enough solution to evaluate the relationship between the query and resources. Yulian (2011) provides basic guidelines to analyze the research gap between the SQR and semantic IR in P2P overlay networks. Doulkeridis *et al*. (2010) provides the solution for the challenges to transform the network from uniform to cluster without global knowledge among the peers in the network. This approach achieves high scalability even under the high rate of content generation because the content is distributed effectively among the peers in the network.

Caching in unstructured P2P search: The effective utilization of the caching in P2P searching schemes reduces the network traffic and response time. A two level semantic caching scheme is introduced in (Garbacki *et al*., 2005). It improves scalability and performance of search by constructing ultra peer architecture on top of the semantic caching. It maintains two level caches in both ultra peer and client peer. Mixed cache-management introduces policy to improve the cache utilization where multiple peers utilize a single file cache at the same time. It achieves high cache hit ratio and reduces cache warm-up period. In Distributed Caching and the Adaptive Search (DiCAS) algorithm (Wang *et al*., 2006), searching divides space into multiple small layers and forwards queries to peers having high query answering capabilities. Adaptive searching forwards queries to the overlay structure of matched peers and thus it reduces the query traffic with shrunk searching space. Distributed caching avoids cache redundancy by efficient utilization of cache space distribution. Consecutively, adaptive search forwards a query to a group of peers and reduces the query traffic with a shrinking searching space.

OVERVIEW OF PROPOSED DYNAMIC CACHING IN SEMANTIC ORIENTED ADAPTIVE SEARCHING (SOAS)

The proposed SOAS approach enhances to facilitate efficient information retrieval by means of semantic indexing through optimized adaptive searching. Self-motivated topology alteration proposed that dynamically adapts to the P2P network topology based on power law distribution. A two-tier flat network constructed over the elected ultra peers in the upper level and the remaining leaf peers are collectively gathered into the lower level (Garbacki *et al*., 2007). Each peer in the network maintains a local cache to store previously retrieved data for requested queries. Query register associated with the ultra peer cache responses of mostly requested queries.

Ultra peer overlay construction using PLD: In the proposed approach, peers are constructed into the loosely coupled unstructured network (Lua *et al*., 2005). This study discusses ultra peer overlay P2P architecture and reveals how to facilitate and enhance the P2P search scheme based on well-organized summary indexing framework.

In P2P overlay construction, each peer runs a Self-Motivated Topology Alteration algorithm to construct an overlay network using power law distribution (Aiello *et al*., 2000). This algorithm not only used to maintain connectivity among the peers in the network and also facilitates to construct the refined topology to achieve efficient search performance. In the power law network, each peer computes its degree from the total number of links (peers) connected into it. Peers having high connectivity (degree) to other peers in the network are termed as high capacity nodes (ultra peers) whereas peers with a low degree are termed as low capacity nodes (client peers). As soon as, low connectivity peers within a certain radius are assembled into a group, it will update its resources and location information to the neighbor peers with high connectivity in the overlay network (Garbacki *et al*., 2007). The aggregation of ultra peers into ultra peer overlay can handle a large number of user generated queries due to its capability. It paved the way for effective query searching where queries are simply flooded into the established overlay networks rather than throughout the peers in the network. The generating function of PLD (Adamic *et al.*, 2001) with 'n' number of peers of 'd' degree vertex connectivity is given by:

$$G[n]=\sum P_d\left(n^d\right)\ 0\le d\le\infty \tag{1}$$

Where:

P_d = The probability for the vertex on the graph
d = The maximum degree of the peer with high connectivity

In case of any failure or breakdown in high connectivity peer, the responsibility of high connectivity peer is handed over to the next high connectivity peer in that group. On that account, generating function of n number of peers is determined as follows:

$$G'[n]=\sum P_d\left(G[n]\right)\ 0\le d\le\infty \tag{2}$$

Symbolic representation: Let k be a set of ultra peer connected to the overlay network. A client peer is represented as $P(I_d, \{O\rightarrow C\}, ID_{SP})$ where I_d is the unique ID of the client peer, $\{O\rightarrow C\}$ represents the set of local files and its corresponding semantic descriptions of the documents stored in the peer and ID_{SP} represents the ID of the ultra peer which connects internally to establish a peer group. The ultra peer is represented as $U(ID_{SP}, C, \{P\}, \{Id_{NSP}\rightarrow C\})$ where ID_{SP} is a unique ID of the ultra peer, C is the semantic descriptions of the documents, {P} is the set of peers constituting the cluster and Id_{NSP} is the set of neighbor ultra peer connected over the ultra peer overlay network.

Publishing information to ultra peer: In the constructed network, each client peer is internally connected to the corresponding ultra peer to advertise its information. In this algorithm, client peers advertise its information to the connected ultra peer through a publishing message as follows:

$$\text{Advertisement}=\{I_d, C\} \tag{3}$$

Where:

AID = The I_d of advertising leaf peer
C = The semantic information maintained by the advertising peer

Therefore, each ultra peer reflects the distribution of content available among the group of client peers connected to it. To improve the efficiency of the search process each ultra peer collects the data summaries of all its associated client peers in the network and summarized content in ultra peer describes the data indexed in its group.

Pre-processing phase: The proposed approach extends classical information retrieval algorithms such as VSM and LSI to represent documents and queries as vectors in the k-dimensional semantic space and quantify the similarity between requested queries and documents. Each peer pre-processes its own local documents independent of other peers in the network and extracts the feature vector from the documents using VSM. LSI is used to derive the index from the extracted vector in order to reduce the high dimensionality space using Singular Value Decomposition (SVD).

Vector space model: VSM is the information retrieval algorithm used to represent semantic data objects as semantic vectors. During the preprocessing phase each peer derives the feature vector from its own local content in order to advertise its content among the ultra peers in the network. Feature vector composes of several elements where each element corresponds to the importance of terms in the document and query. The feature vector is represented as:

$$FV=\{FT, W, TFID\,(l, m), N\} \tag{4}$$

Where:

FT = A feature term
W = Term weight
N = A number of documents TFID (l, m) denotes that the lth coordinate of the mth transformed document and it is represented as:

$$TFID(l, m)=TF(l, m)*IDF(l)$$

In TF·IDF Term-Weighting Method, weight of the element in the vector is computed using the statistical formula W = TF×IDF. It assigns weight according to the occurrence of words frequently in the document as weighted term frequencies, namely Term Frequency (TF) and Inverse Document Frequency (IDF) using zipfian distribution. TF denotes a number of times the term appears frequently in the document and also the crucial factor to discriminate one document from other documents. Inverse IDF represents the number of times the term appears in the other documents (Salton and Buckley, 1988).

Latent semantic indexing: Literally, VSM uses semantic vectors to retrieve a data object but it suffers from polysemy (the same word with a different meaning), synonymy (different word with the same meaning) and noise problems. In order to overcome the problems of information retrieval, LSI statistically derives indices from summarized semantic vectors. In P2P, a large number of peers are available in the network, each peer contains a large volume of the document to share with other peers in the network. Therefore, a large semantic space is occupied by documents with high dimensionality. Singular Value Decomposition (SVD) is used to convert high-dimensional term vector (computed from VSM) into lower-dimensional semantic vector. Information is retrieved from the location of the content merely through LSI (Deerwester *et al*., 1990) where local index is derived in accordance with similarities exists between semantically related documents in dimensionality semantic space. Hence, the proposed scheme derived local indices of peers are distributed among ultra peers in an overlay network that ultimately improves searching efficiency at low processing cost.

Proposed algorithm: n is a number of peers, m is a number of ultra peers, peers {P_1, P_2, ..., P_n}, ultra peers {UP_1, UP_2, ..., UP_m}, semantic data objects {S_{D1}, S_{D2} ..., S_{dn}}, semantic vectors {S_{V1}, S_{v2}, ..., S_{vm}} and local index {S_{I1}, L_{I2}, ..., L_{im}}. The study shows hierarchical summarizing and indexing. The algorithm of summarizing and indexing performs the following steps.

Step 1 (Semantic vector transformation): Vector space odel summarizes the semantic vectors according to the similarity between terms of requested query and semantic mobjects in the document. Semantic vector is represented in the form of large document matrix M_{ij} where rows represent terms and column represents the documents.

Step 2 (Dimensionality reduction): Singular Value Decomposition (SVD) converts the term document matrix of dimensionality into three matrices as U, S and V.

Step 3 (Index derivation): Original matrix M is accomplished through rearranging three metrics as follows:

$$U*S*V = M$$

The Latent Semantic Indexing is derived from the column vectors of the reduced M matrix. Finally, a derived index is associated with its corresponding documents.

Hierarchical summarizing and indexing:

```
Step 1: Semantic vector transformation
  For each peer {m<n}
        for (i = 1, i< = n, i++) {
         $S_V[i]$→VectorSpacee Model {$S_d[i]$}
         $S_V[i]$ = {FT, W, TFIDF (l, m), N}
         Generate group and global weighted term
frequency;
        }
Step 2: Dimensionality reduction
          Low                         Dimensionality
{($S_{VL}[i]$)}→$S_{VD}${(High dim ensionalit) y($S_V[i]$)}
Step 3: Index derivation
          for (i = 1, i< = n, i++) {
$L_I[i]$→Latent Semantic Indexing {$S_{VL}[i]$}
  for [j = 1; j = m; j++]
     UPLOAD $L_I[i]$ to its ultra peer UP[j]}
```

Summarizing and indexing: The proposed P2P searching employs multi-level hierarchical summary indexing framework to enhance the quality of search results through optimized searching. In order to attain efficient and optimized searching, queries are only forwarded to peers having high answering capabilities which are determined from derived indices possessed by each peer in the network. In peer level, each peer derives Local Index (LI) from the summarized information. In peer cache level, each peer derives the Cache Index (CI) from its cached responses of previously requested queries. In query register level, Query Register (QR) derives the Response Index (RI) from cached responses of previously requested queries maintained in QR. In ultra peer level, it derives indices such as Global Index (GI) which is derived from information maintained among leaf peers in its peer group.

Query processing: The previous study clearly provides the techniques to derive the index from semantic objects. In order to support efficient query processing, derived indices are maintained as an appropriate internal structure to store and process the dynamic collection of indices using operations using suitable operation evaluation algorithm (Michal *et al*., 2007). The proposed SOAS leads to restrict in small searching scope with the finite number of ultra peers in the network. In the proposed context, query processing is first proceeded through determining which ultra peer possess required relevant data from the summarized information using summary and index hierarchy searching. If a source peer requires file, it issues a query request with additional semantic information. In this approach, generated queries clearly define conceptually related partial information, so the query request itself provides the minimum information about the required results. The query request is in the form as follows:

$$\text{Query Request}\{I_d, \text{RFN}, \text{C}, \text{TTL}\} \tag{5}$$

Where:

I_d = The peer ID that generates the query
RFN = The name of the required file which is not mandatory
C = The query string
TTL = A query indicates the maximum tolerable size

Rounds of query searching: When the source peer generates a query request, the feature vector is derived from the concepts and then content based query searching processing is performed according to the relevancy between the semantic vectors. The indexes maintained among the ultra peers which is used to locate the results of the destination peers. In order to improve the accuracy of the results, multiple query lookups are performed among the peers in the network. A query request may result in multiple lookups in order to improve accuracy. Figure 1 depicts five rounds of SOAS. Query processing is performed as follows:

First round search: Query is searched in its own local files using Local Index (L_I). If a strong semantic similarity exists between two vectors then corresponding document of semantic vector returned to the requested user and local hit will occur (Doulkeridis *et al*., 2007). Otherwise, invokes the next round search:

$$\text{REL}(\text{Q, Local files}) = \sum \text{QI} * \text{LI} \tag{6}$$

Where:

QI = Query Index

LI = Local Index of local files

Second round search: Query is searched in its own local cache using Cache Index (CI). If the local cache hit occurred, the requested data returned to the source peer. Otherwise, forwards the query to corresponding ultra peer then invokes the third round search:

$$\text{REL}(\text{Q, Local cache}) = \sum \text{QI} * \text{CI} \tag{7}$$

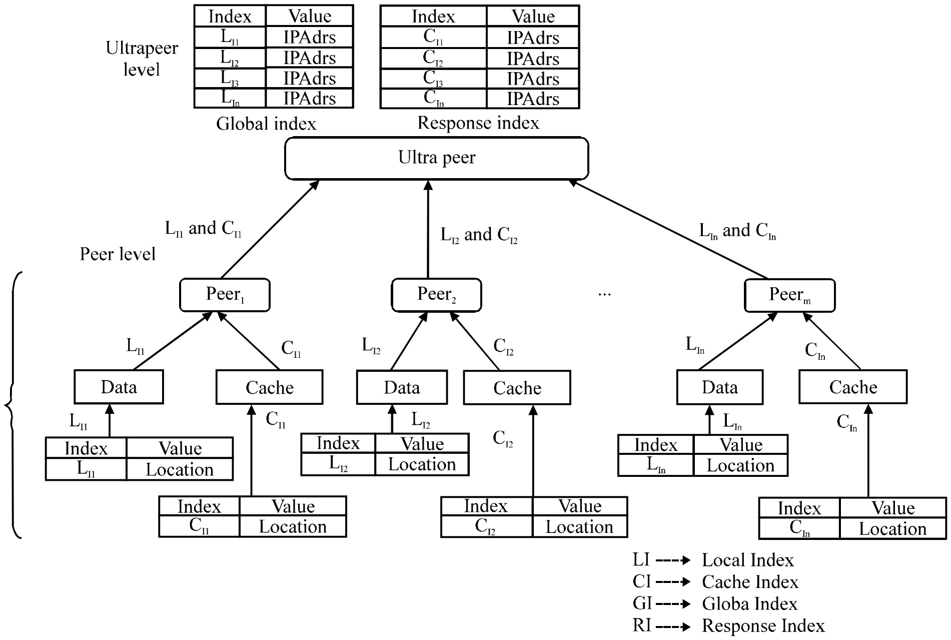
where, CI is the Cache Index for data item in the local cache.

Third round search: Query is forwarded to its own ultra peer UP_I where query searching is processed through response index which is derived from the responses of frequently requested queries in query register. For a high relevance score, queries register returns required resource to source peer along the requested path. If it fails, the fourth round search is started. Noticeably, relevance score is computed from requested query and content of the query register as follows:

$$\text{REL}(\text{Query, Register}) = \sum \text{QI} * \text{RI} \tag{8}$$

where, RI is Response Index derived from most requested queries.

Fourth round search: Query search begins using Global Index (GI) (an aggregation of local index), GI is derived from the data items available among the client peers connected to the ultra peer. If the higher degree of relevance exists between the query and document then required resource is returned to the source. If it fails then the fifth round search is started:



Index	Value
L_{I1}	IPAdrs
L_{I2}	IPAdrs
L_{I3}	IPAdrs
L_{In}	IPAdrs

Index	Value
C_{I1}	IPAdrs
C_{I2}	IPAdrs
C_{I3}	IPAdrs
C_{In}	IPAdrs

Index	Value
L_{I1}	Location

Index	Value
C_{I1}	Location

Index	Value
L_{I2}	Location

Index	Value
C_{I2}	Location

Index	Value
L_{In}	Location

Index	Value
C_{In}	Location

Fig. 1: Five-round semantic oriented adaptive search

$$\text{REL}(\text{Q, Ultrapeer}) = \sum \text{QI} * \text{GI} \tag{9}$$

where, GI is a Global Index.

Fifth round search: If the requested data item is not available in its forwarded ultra peer, a fifth round search is invoked where the query is forwarded to neighbor ultra peer using knowledge based adaptive searching. Therefore, requested query can be satisfied by any one of the neighbor ultra peers in the overlay network.

Knowledge based adaptive searching: In the fifth round of search, knowledge based adaptive search is proposed to forward query among the ultra peers in the ultra peer overlay network. Every peer computes the probabilistic function using knowledge learned from the experience regarding to query hit, searching time and local topological information. In query processing, queries are forwarded based on an appropriate decision taken intelligently based on the affiliation between Hop Count (HC) of query message and decision Threshold (T) using constructed probabilistic function. Query processing is performed in two stages called as short-term search (Directed flooding) and long-term search (Biased Random Walk) or Depth First Searching (DFS).

Short term search when HC≤T: TTL is maintained in each query message represents as hop count. If the hop count is less than the decision threshold then queries are forwarded among ultra peers using flooding or Breadth First Searching (BFS) where the query is flooded among peers in a defined hop count. In directed flooding, peer forwards the query message probabilistically to a subset of neighbors that has the ability to provide more high-quality search results than other neighbor peers in the network. At this point, a subset of neighbors to forward a query is determined by the product of degree of connectivity d and transmission probability p:

$$\text{Subset of neighbors to flood query} = d * p \tag{10}$$

If p = 1 then normal flooding takes place, otherwise directed flooding takes place that forwards the query message to a subset of neighbors (d*p). If a hop count is 2 then query message is flooded among peers in two hops.

Long term search when HC>T: If a hop count is greater than the decision threshold, the biased random walk strategy is used to flood the query among peers in the network. It resembles that instead of forwarding queries to randomly chosen neighbors; each peer forwards the queries through taking an appropriate decision from the knowledge learned from the previous experience to a selected neighbor has a high probability to provide high quality. SOAS algrithim shows step by step process of SOAS algorithm. Requested Peer (CP), Requested Query (Q), Required File Name (RFN), TTL (HC) is hop count, threshold (T) is 3, Query register (QR).

SOAS algorithm:

```
Process SOAS ($I_d$, RFN, C, TTL)
Client peer generates query Q
 Step 1: //LI checks R is available in Local Files
        do {
              Compute REL (Q, Local   Files) = $\Sigma$ L1*Q1
              If REL (Q, Local Files)>threshold      {
              Then Return Required File→  CP;
              }
         }
 Step 2:  //CI checks R is available in Local Cache
        else do {
             Compute REL (R, Local Chace) = $\Sigma$ CI*Q1I
             If REL (Q, Local Chace)>threshold    {
             Then Return Required File → CP;
             }
    }
 Step 3: //RI checks R is available in Query Register
        else do {
               Forward Q from CP→OWN Ultrapeer
               Compute REL (R, QR) = $\Sigma$ RI*QI
                If REL (Q, QR)>threshold   {
                     Then Return Required File →CP;
                }
         }
 Step 4: //GI checks R is available in its peer group
         else do {
                Compute REL (Q, QR) = $\Sigma$ G1*Q1
                If REL(Q, QR)>threshold      {
                 Then Return Required File → CP;
                }
        }
 Step 5: else do {
        Knowledge based Adaptive Searching
        }
```

Dynamic TTL based caching: In the proposed SOAS, dynamic TTL based caching deployed as a two level caching where both ultra peer and client peer in the returning path can intermittently store the responses for previously requested queries that provide enough guidance for future message routing (Bobby *et al*., 2003). In the ultra peer cache, responses of recently requested queries by the client peers are maintained whereas client peers maintain the cache to store the responses of previously generated queries. The main advantage of this cache arrangement is to utilize the available storage efficiently to reduce query latency by retrieving responses for previously requested queries directly from cache which possess matched responses among the peers. In addition, cached information in a low degree peer must be properly updated to its connected peer with a high degree connectivity in the network of zipf distribution (Lee *et al.*, 1999) because query forwarding is carried out only to the overlay network. A cache entry consists of 3 fields: Index, cached information and a timestamp (TTL). The TTL associated with the cached data is used to determine that whether cached data is true or stale. When the TTL of cached data becomes zero, the state of the data is set to stale.

The proposed dynamic TTL based caching algorithm typically performs effective cache management which is more beneficial for the caching of the dynamic contents among the peers in Peer to Peer (P2P) networks. Cache associated with each peer is refreshed dynamically according to the query request rate (popularity) and TTL. The popularity of the cached file is determined from the number of times which is mostly requested by the peers in the network. If the cache becomes full, a more efficient Cache Replacement Algorithm should keep the popular files which were the responses of mostly requested queries. The least popular file with less request rate is removed from the cache. In this approach, each cached datais assigned with TTL (expiration time) beyond which the cache stops to serve the new request until it is validated. The principal fixation is that responses of most requested queries are updated with high TTL rather than responses of rarely requested queries. Data replication is also efficiently handled by query registry to obtain the minimum average delay. In this approach, there is no need to survive for searching to locate its original location whereas location of a data item is easily retrieved without convoluted searching and reduces network traffic with high cache hit ratio. The proposed approach not only improves search performance, it also improves the freshness of content sharing under TTL-based consistency.

PERFORMANCE EVALUATION

In this study, the evaluation of the proposed approach is carried out in two phases. In the first phase, evaluation of the proposed approach is carried out in a small scale real P2P network with 100 peers. Whereas in the second phase, peerSim simulator is used to simulate the SOAS searching scheme to illustrate that proposed hierarchical summary indexing framework to achieve efficiency and scalability in a large scale P2P System.

P2P network environmental setup: In this study, researchers built a semantic overlay network using the power law distribution in order to validate the proposed SOAS approach. Using JXTA protocol (Li, 2001), researchers construct a small scale real P2P network with 100 peers and then evaluate the approach using CMU Text learning group data set consists of 20,000 messages collected from 20 newsgroups (Lang, 2012). Each group consists of 1000 documents related to electronics, hardware, sports, politics, religion, etc., In order to validate the approach, we assign 200 documents to each peer and the total number of peers available in the network is 100. A hierarchical summary and indexing framework for ultra peer architecture is deployed where VSM is used to generate a term by document matrix. SVDPACK package is used to apply SVD to reduce the dimensionality of semantic space. LSI derives the index for the semantic vector in 1.7 GHz Pentium IV machine with 1 GB of memory. The number of queries used to evaluate performance is 1000. It returns relevant results related to user generated queries.

In the second phases of evaluation, the proposed approach is validated using peersim simulator in a large scale network consists of 10000 peers. The simulation model is executed for 10,000 queries. The TTL of the queries is set at 50. In this evaluation, experimental analysis is performed to evaluate the effectiveness of the proposed mechanism. Each peer in the P2P network is composed of arbitrarily initialized routing tables. In simulation, each peer in the network creates the new and different queries autonomously. A peer creates a query at the rate of 0.005 sec for each step in the simulation. A peer in the network may satisfy the recently created query with equal probability.

Experimental results: In order to determine the performance of the proposed SOAS approach, various performance evaluation metrics are calculated based on the relevant dataset. The experiment was carried out to compare the performance of the proposed SOAS scheme with an existing Semantic Content Based Searching (SCBS).

Accuracy of returned results

Precision (P): Precision is a measure to determine how well the approach can perform to discover only relevant data items.

Recall: Recall is a measure to determine the ability of the approach to discover all relevant data items from the available data item among the peers in the network.

Figure 2 illustrates the resulting average precision-recall curves for the proposed approach. The proposed SOAS approach the index is derived from semantic data objects. Consequently, it leads to return high quality search results through full fledged query searching.

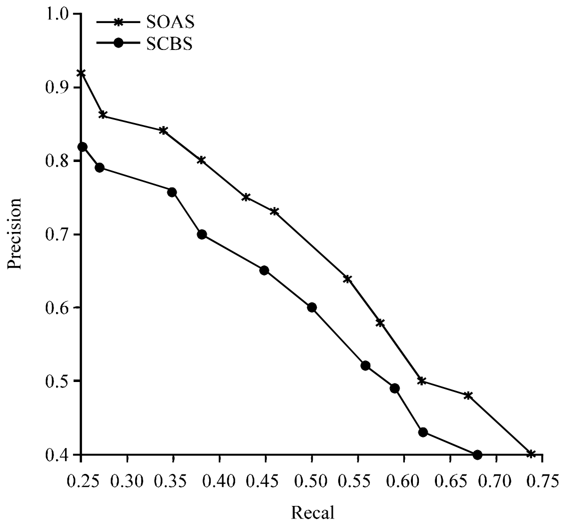


Fig. 2: Accuracy of returned results

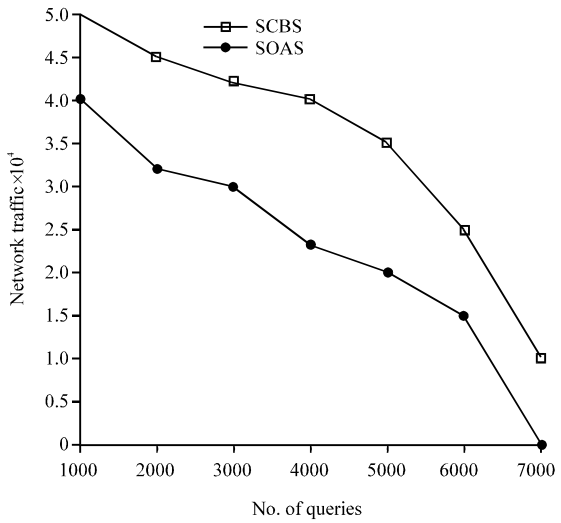


Fig. 3: Average network traffic of SOAS

Semantic similarity is found through cosine similarity between the query vector and semantic vector of client peers. Therefore, the proposed approach returns high relevant results as per relevant dataset.

Network traffic: It represents the total amount of traffic generated during the queries processing in the network divided by the number of queries generated by peers in the network. T_i is the total amount of traffic generated by queries and Q_i is the number of queries generated by peers. In the proposed SOAS scheme, undesirable network traffic is reduced through three enrich tasks such as leaf peer advertises an index to ultra peer through a single message, limited searching scope in ultra peer overlay rather than flooding the query throughout the network and query results for already requested queries are easily retrieved from the query register or peer cache. Figure 3 illustrates that in the proposed SOAS, network traffic is significantly reduced according to the increasing number of queries when compared to a SBCS searching scheme.

Response time: It is defined as the time taken to forward the query reply to the requested peer. In the proposed SOAS scheme, query register associated with ultra peer can dynamically cache the responses of requested queries according to the query request rate. Therefore, the response time of already requested query is considerably reduced because it can be retrieved directly from the cached responses in the query register. Researchers can easily reveal that the number of queries requested from peers increases the number of cached data items. Hence, obviously increasing the number of cached data item meets queries requested from other peers in the network. Figure 4 shows the proposed scheme. It takes nearly 15% lesser hops to forward the reply than the SCBS searching scheme.

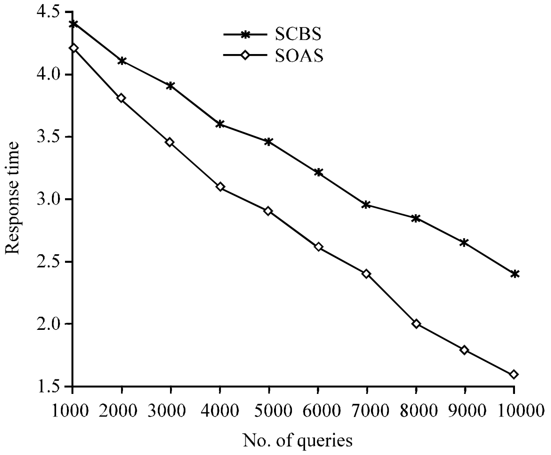


Fig. 4: Average response time of SOAS

Cache hit ratio: It is defined as the ratio of requests queries satisfied through cached information in the cache rather than broadcasting a query message. It is represented as the number of messages routed through cached content out of 100 messages. It is a crucial factor to improve the searching efficiency with high success rate and less response time.

C_i is the number of successful searches routed through a cache and Q_i is the number of queries generated

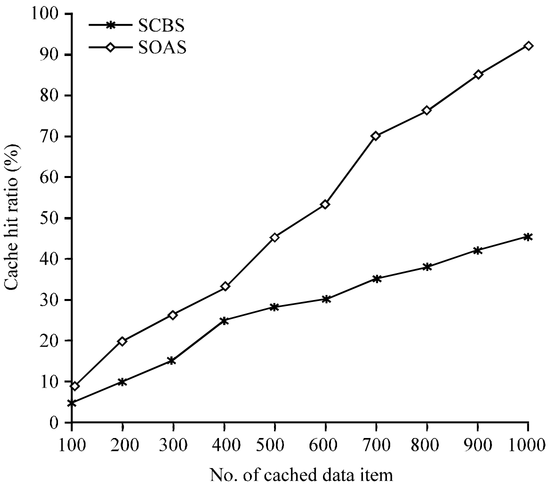


Fig. 5: Cache hit ratio of SOAS

by peer "i". In the proposed scheme, cache is maintained in each peer and so ultra peer also updates the indices of cached data item in its peer group. The high cache hit ratio can be obtained through dynamically caching of a large number of responses from previously requested queries. Figure 5 reveals that the proposed scheme achieves almost 92% of cache hit ratio. Hence, it is concluded that a high cache hit ratio is obtained through the proper notification of the cached response among peers with its connected ultra peer.

Success rate: Success rate is the number of queries that successfully generated the responses. It explicitly relies on how query processing efficiently perform requested queries. S_i is the number of successful searches done by peer "i" and Q_i is the number of queries generated by peer "i". The key aspect is to improve the success rate using indexing of data item in each peer. In the proposed SOAS scheme, ultra peer overlay is constructed efficiently and also achieves fast convergence due to data information retrieval through an indexing framework where no need to flood the query throughout the network for requested query.

The most serious consideration to achieve a high success rate is that queries which does not satisfy by its own peer group, queries are searched in neighbor ultra peer using adaptive searching.

Figure 6 shows that the existing SCBS achieves 74% of success rate but the proposed scheme achieves 88% of success rate. This is due to the dynamic caching of responses according to the query request rate and limited searching scope.

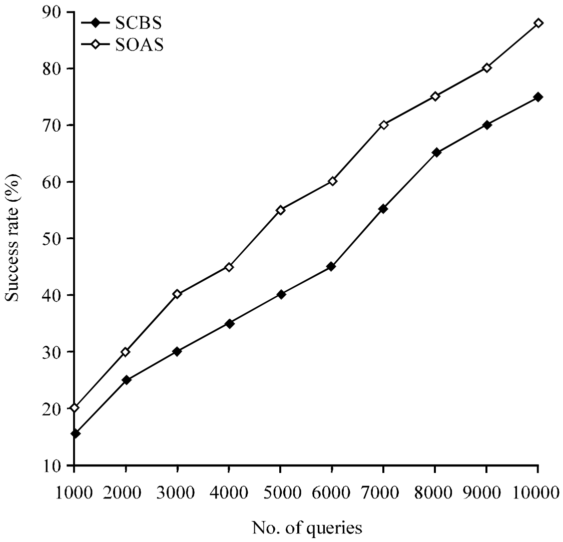


Fig. 6: Average success rate of SOAS

CONCLUSION

In this study, an extensible two-tier flat network structure is constructed with an efficient summary indexing framework. It also addresses the issues regarding content based searching in the P2P network. Based on this framework, researchers have proposed a five-level search based on summary indexing derived from semantic objects. A vector space model is used to transform large size semantic objects into small high-dimensional vectors. The Latent Semantic Indexing is used to derive the index from the transformed high dimensional points. The derived indices are used to perform efficient peer and document search on the corresponding level. The proposed SOAS search significantly reduces the searching network traffic by maintaining an index in ultra peer and search is limited in a smaller area. In addition, the query register associated with ultra peer can dynamically cache the response of previously requested queries. In order to achieve efficient cache management, a dynamic TTL based cache consistency is proposed to retain the fresh data based on the query popularity rate and TTL. Therefore, searching efficiency is improved with reduced response time and network traffic. The simulation results prove that the proposed research is effective. From the simulation, it is clear that the summary indexing structure is adaptive and achieves better performance.

REFERENCES

Adamic, L.A., R.M. Lukose, A.R. Puniyani and B.A. Huberman, 2001. Search in power-law networks. Phys. Rev. E, Vol. 64. 10.1103/PhysRevE.64.046135.

Aiello, W., F. Chung and L. Lu, 2000. A random graph model for massive graphs. Proceedings of the 32nd Annual ACM Symposium on Theory of Computing Portland, OR, USA., May 21-23, 2000, ACM, New York, USA., pp: 171-180.

Bobby, B., S. Chawathe, V. Gopalakrishnan, P. Keleher and B. Silaghi, 2003. Efficient peer to peer searches using result-caching. Lect. Notes Comput. Sci., 2735: 225-236.

Chen, W. and L. Xiao, 2007. An effective P2P search scheme to exploit file sharing heterogeneity. IEEE Trans. Parallel Distrib. Syst., 18: 145-157.

Crespo, A. and H. Garcia-Molina, 2005. Semantic overlay networks for P2P systems. Proceedings of the 3rd International Workshop on Agents and Peer to Peer Computing, July 19, 2005, New York, USA., pp: 1-13.

Deerwester, S., S.T. Umais, G.W. Furnas, T.K. Landauer and R. Harshman, 1990. Indexing by latent semantic analysis. J. Soc. Inform. Sci., 41: 391-407.

Dimitrios, T. and N. Roussopoulos, 2003. Adaptive Probabilistic Search (APS) for Peer to Peer Networks, Proceedings of the 3rd International Conference on Peer to Peer Computing, September 1-3, 2003, University of Maryland, pp: 102-109.

Doulkeridis, C., A. Vlachou, K. Norvag, Y. Kotidis and M. Vazirgiannis, 2010. Efficient search based on content similarity over self-organizing P2P networks. Peer to Peer Network. Applic., 3: 67-79.

Doulkeridis, C., A. Vlachou, Y. Kotidis and M. Vazirgiannis, 2007. Peer to peer similarity search in metric spaces. Proceedings of the 33rd International Conference on Very Large Data Bases, September 23-28, 2007, ACM, New York, pp: 986-997.

Garbacki, P., D.H.J. Epema and M. van Steen, 2005. A two-level semantic caching scheme for super-peer networks. Proceedings of the 10th International Workshop on Web Content Caching and Distribution, September 12-13, 2005, Sophia Antipolis, France, pp: 47-55.

Garbacki, P., D.H.J. Epema and M. van Steen, 2007. Optimizing peer relationships in a super-peer network. Proceedings of the 27th International Conference on Distributed Computing Systems, June 25-27, 2007, Toronto, Canada, pp: 31.

Jiang, S., L. Guo, X. Zhang and H. Wang, 2008. Light flood: Minimizing redundant messages and maximizing scope of peer-to-peer search. IEEE. Trans. Parallel Distrib. Syst., 19: 601-614.

Kalogeraki, V., D. Gunopulos and D. Zeinalipour-Yazti, 2002. A local search mechanism for peer to peer networks. Proceedings of the International Conference on Information and Knowledge Management, November 4-9, 2002, ACM, New York, USA., pp: 300-307.

Lang, K., 2012. CMU text learning group data archives. http://www.cs.cmu.edu/afs/cs.cmu.edu/project/theo-20/www/data/news20.html.

Lee, B., P. Cao, L. Fan, G. Phillips and S. Shenker, 1999. Web caching and Zipf-like distributions-evidence and implications. Proceedings of 18th Annual Joint Conference of the IEEE Computer and Communications Societies, March 21-25, 1999, Palo Alto Reserch Center, USA., pp: 126-134.

Li, G., 2001. Project JXTA: A technology overview. Technical Report, Sun Microsystems.

Lu, J. and J. Callan, 2007. Content-based peer to peer network overlay for full-text federated search. Proceedings of 8th RIAO Conference on Large-Scale Semantic Access to Content (Text, Image, Video and Sound), May 30-June 01, 2007 Pittsburgh, Pennsylvania, pp: 490-509.

Lua, E.K., J. Crowcroft, M. Pias, R. Sharma and S. Lim, 2005. A survey and comparison of peer-to-peer overlay network schemes. IEEE Commun. Surv. Tutor., 7: 72-93.

Mei, L., L. Wang-Chien, A. Sivasubramaniam and J. Zhao, 2011. SSW: A small-world-based overlay for peer-to-peer search. IEEE Trans. Knowl. Data Engin., 19: 735-749.

Michal, B., D. Novak and P. Zezula, 2007. MESSIF-metric similarity search implementation framework. Res. Dev., 4877: 1-10.

Milojicic, D., V. Kalogeraki, R. Lukose, K. Nagaraja and J. Pruyne *et al*., 2003. Peer to peer computing. Technical Report, Hewlett-Packard.

Nejdl, W., M. Wolpers, W. Siberski, C. Schmitz, M. Schlosser, I. Brunkhorst and A. Loser, 2003. Super-peer-based routing strategies for RDF-based peer to peer networks. Web Semantics, 1: 177-186.

Salton, G. and C. Buckley, 1988. Term-weighting approaches in automatic text retrieval. Inform. Process. Manage. Int. J., 24: 513-523.

Shen, H.T., Y. Shu and B. Yu, 2004. Efficient semantic-based content search in P2P network. IEEE Trans. Knowl. Data Engin., 16: 813-826.

Tsoumakos, D. and N. Roussopoulos, 2003. A comparison of peer to peer search methods. Proceedings of the 6th International Workshop on Web and Databases, June 12-13, 2003, San Diego, California, pp: 61-66.

Tsungnan, L., P. Lin, H. Wang and C. Chen, 2009. Dynamic search algorithm in unstructured peer to peer networks. IEEE Trans. Parallel Distrib. Syst., 20: 654-666.

Wang, C., L. Xiao, Y.H. Liu and P. Zheng, 2006. DiCAS: An efficient distributed caching mechanism for P2P systems. IEEE Trans. Parallel Distrib. Syst., 17: 1097-1109.

Wu, C.J., K.H. Yang and J.M. Ho, 2006. AntSearch: An ant search algorithm in unstructured peer to peer networks. Proceedings of the 11th IEEE Symposium on Computers and Communications, June 26-29, 2006, IEEE Computer Society Washington, DC, USA., pp: 429-434.

Yingwu, Z. and Y. Hu, 2006. Enhancing search performance on gnutella-like P2P systems. IEEE Trans. Parallel Distrib. Syst., 17: 1482-1495.

Yulian, Y., 2011. Semantic information retrieval over P2P network, CORIA 2011. Proceedings of 8th French Information Retrieval Conference, March 16-18, 2011, Universitaires d'Avignon, pp: 391-396.