Cluster-based Intelligent Searching in Unstructured Peer-to-Peer Networks [*]

Xiuqi Li and Jie Wu
Department of Computer Science and Engineering
Florida Atlantic University
Boca Raton, Florida 33431
{xli, jie}@cse.fau.edu

Abstract

Existing cluster-based searching schemes in unstructured peer-to-peer (P2P) networks employ flooding/random forwarding on connected dominating sets (CDS) of networks. There exists no upper bound on the size of CDS of a network. Both flooding and CDS hinder query efficiency. Random forwarding worsens the recall ratio. In this paper, we propose a cluster-based searching scheme that intelligently forward queries on the maximum independent sets (MIS) of networks. Our approach partitions the entire network into disjoint clusters with one clusterhead (CH) per cluster. CHs form a MIS and are connected through gateway nodes. Each node takes one role, a CH, a gateway, or an ordinary node. A CH looks up the data for the entire cluster using data summaries of cluster members, which are represented by bloom filters. Between clusters, CHs intelligently forward queries via gateways to the best neighbor CHs that are most likely to return query results. The experimental results demonstrate that our scheme greatly improves the query efficiency without degrading the quality of the query results, compared to existing approaches.

1. Introduction

Existing searching schemes in unstructured P2P networks can be categorized as regular-grained or coarse-grained based on whether all nodes perform the query forwarding [1]. Coarse-grained methods are better than regular-grained methods in that they reduce the query load, increases the scalability, and supports node heterogeneity. However, in current coarse-grained methods [2] [3] , queries are flooded or randomly forwarded between clusters via a *connected dominating set (CDS)* of a network. A CDS is a connected subset where each node in the network is either in the set or a neighbor of a node in the set. Flooding causes many duplicate query messages. There exists no upper bound on the sizes of the CDS of networks. Both hamper the query efficiency. Random forwarding degrades the quality of query results. It is also observed that in the dominating-set-based searching, a node may be dominated by two different connected dominating nodes and is unnecessarily checked twice when processing the same query.

To alleviate these problems, we propose a cluster-based intelligent searching scheme in unstructured P2P networks that employs a *maximum independent set (MIS)*. A MIS of a network is a subset of that network where nodes in this subset are not neighbors of each other and each node in the rest of the network is a neighbor of some node in this subset. Our approach first divides the entire network into disjoint clusters each of which has a clusterhead (CH), which are connected through gateways. All CHs form the MIS of the P2P overlay. A node that is neither a CH nor a gateway is an ordinary node. During query processing, CHs handle the data lookup within their own clusters using data summaries about their cluster members, which are generated using bloom filters [5]. CHs also intelligently forward queries to neighbor CHs that are most likely to return query results.

It is known that there exists an upper bound on the size of the MIS of a network. By employing MIS instead of CDS, our approach achieves higher query efficiency than existing methods, such as dominating-set-based searching. In addition, our approach improves the quality of query results by utilizing intelligent query forwarding between non-overlapping clusters. To the best of our knowledge, no existing work exploits both cluster-based intelligent searching and MIS. The experimental results demonstrate the superior performance of our approach.

This paper is organized as follows. In Section 2, existing related work is reviewed. In Section 3, the proposed approach is described in detail. The discussion focuses on the query processing, the clustering strategy, the data summary

[*] This work was supported in part by NSF grants CCR 0329741, CNS 0422762, CNS 0434533, and EIA 0130806. Contacting address: jie@cse.fau.edu

within clusters and the neighbor CH profiles. In Section 4, experimental results are presented. At the end, the paper is summarized and future work is identified.

2. Related work

Existing search schemes in unstructured P2P networks can be regular-grained or coarse-grained. In a regular-grained approach, all nodes participate in query forwarding. In a coarse-grained scheme such as dominating-set-based searching [2] and the ultrapeer approach [3], only a subset of nodes perform query forwarding. The dominating-set-based searching first finds a CDS of the P2P network using the extended marking process. Then a random walk is conducted on the CDS. The searching terminates with a maximum TTL. The ultrapeer approach [3] classifies nodes into ultrapeers and leafs during node joins. Each leaf maintains only one single connection, which has to be a connection to an ultrapeer. The ultrapeers form a CDS, not a MIS. Queries are flooded on the set of ultrapeers. In an extension of the ultrapeer approach [4], each ultrapeer uses a bloom filter to keep data on a leaf neighbor. It is similar to the work in this paper. However, the searching is still flooding on a CDS.

3. The cluster-based intelligent searching

Figure 1 illustrates the proposed cluster-based intelligent searching (CIS) algorithm. Documents are indexed by keywords. A query is a set of keywords. The query result is the set of documents indexed by all the keywords in the query. We distinguish between searching and downloading. The CIS algorithm locates the desired files. Users determine whether and where to download the found files. Figure 1(a) is a simple example of an unstructured file sharing P2P. The keywords used for indexing documents on each node are indicated next to the node id. The CIS algorithm first divides the network into disjoint clusters shown in Figure 1(b). Each cluster has an elected *clusterhead (CH)*. CHs communicate with their neighbor CHs via *Gateways*. Two CHs are neighbors of each other if their distance is at most 3 hops.

The searching is carried out among CHs. Within a cluster, the CH looks up data for itself and on behalf of all its non-CH members. If a non-CH node issues a query, it asks its CH to process that query. Between clusters, each CH intelligently forwards received unique queries to the top J neighbor CHs that are most likely to return query results based on the past queries answered via those neighbor CHs. Gateways only forward queries to the intended CHs indicated by the forwarding CHs. Ordinary nodes do not participate in the query traffic. Redundant queries are detected using unique query IDs and are discarded at the receiving nodes. A query path terminates when the query results are found or when the TTL expires. Each CH keeps a data summary about the keywords available on each of its non-CH member. To reduce the storage space, the data summary is stored in a compressed form using a bloom filter. Each CH also maintains a profile for each of its neighbor CHs, which stores the past queries answered via that neighbor CH.

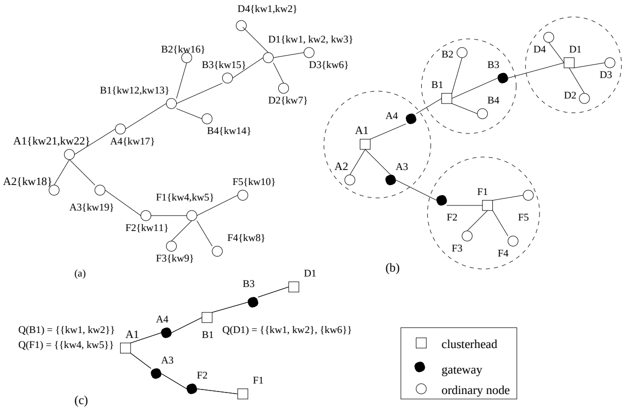


Figure 1. An overview of the cluster-based intelligent search. (a) The P2P overlay and keyword distribution. (b) the clustered P2P overlay. (c) Nodes that are actually involved in the searching.

For example, in Figure 1(c), A_1 wants documents that contain keywords, KW_1, KW_2, and KW_3, which are not in its own cluster. A_1 knows that a query for documents that contains the keywords KW_1 and KW_2 was answered via the neighbor CH B_1. A query for documents that contains the keywords KW_4, and KW_5 was answered via the neighbor CH F_1. A_1 then decides that it is more likely to get query results via $B1$ than via F_1 because similar documents were found via B_1. The query is then forwarded to B_1 via A_4. Similarly, B_1 knows that a similar query for documents that contain keywords KW_1 and KW_2 were found via its neighbor CH D_1. It then forwards the query to D_1 via B_3. D_1 found the desired documents on D_3's data store using the data summary. The query results are returned to $A1$ via the reverse query path.

A similar approach was proposed in [9]. However, it is designed for a non-clustered unstructured P2P. Therefore, it does not consider issues related to clustered unstructured P2Ps. The major issues include how to handle the searching within one cluster, how to track the query results returned from a particular neighbor CH given the fact that CHs are not direct neighbors of each other, and whether the gateways should perform intelligent query forwarding. Our approach successfully resolves these issues.

3.1. The clustering strategy

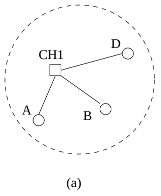
First, CHs are elected distributedly. Then each CH determines its gateways to its neighbor CHs. The CH election favors primarily nodes with high degrees and secondarily nodes with the ability to answer many queries. It is assumed that the more documents and keywords available on nodes, the more queries that may be answered by those nodes. A CH and the selected gateways to its neighbor CHs form a 3-level tree that is rooted at that CH. These gateways are selected branch by branch until all neighbor CHs are covered (reached). First the best branch that covers the most neighbor CHs is chosen, then the second best. Our gateway selection is adapted from the idea for ad-hoc networks in [6]. The adaptation includes proactive instead of on-demand gateway selection and 3-hop instead of 2.5-hop neighborhood.

Cluster membership maintenance includes node joins and leaves/failures. A new node first connect to existing nodes of any kind and then selects its CH from its neighbors. When a node leaves/fails, nodes in its 3-hop neighborhood removes it from their own neighborhoods. Additional actions, such as updating the neighbor CH sets and/or associated gateways and selecting new CHs, may be performed depending on the role of the leaving node.

3.2. The data summaries within clusters

The data summary of a cluster member is represented as a bloom filter [5], a data structure typically used as an approximate representation of a set of n elements. It is a w-bit wide bit string that is equipped with h number of hash functions. There are false positives but no false negatives when using bloom filters. A false positive indicates the existence of a data piece in a set by a bloom filter, but the data piece actually does not belong to that set.

We follow the guideline in [7]: when w/n = 6, and h = 4, the error rate is 0.0561. The four hash functions are generated using SHA-1, which is chosen for its well-known properties. The bit pattern of a keyword is computed as follows. First use the SHA-1 hash function to hash the keyword to a 160-bit number. Then break this number into four 40-bit numbers and take the modulus of each 40-bit number by the bloom filter width w. The four resulting numbers are used as the results of the four hash functions. Multiple keywords may be hashed to the same bit position. This is handled by keeping a counter for each bit in a data summary. The data summary is created by cluster members and sent to their CHs. An example of the data summaries is illustrated in Figure 2. The data summary for node A is the bloom filter, 110010001010001. Therefore, CH_1 knows that documents indexed by the keywords, KW_6 and/or KW_7 may be found on node A.



Node	Keywords available
A	{KW6, KW7}
B	{KW1, KW2, KW3}
C	{KW4, KW5}

(b)

Node	Data Summary
A	1 1 0 0 1 0 0 0 1 0 1 0 0 0 1
B	1 0 0 1 1 0 0 1 0 0 1 0 1 1 0
C	0 1 1 0 0 1 1 0 0 1 0 1 0 0 1

(c)

Figure 2. (a) The cluster with 4 nodes, the clusterhead CH_1 and its members A, B, and D. (b) The keywords available on each cluster member. (c) The data summaries kept at the clusterhead CH_1.

3.3. Neighbor clusterhead profiles

The profile for a neighbor CH c includes the most recent queries whose results were returned via c, the total number of documents in the entire cluster of c, the total number of keywords in the entire cluster of c, and the path to c. When the past performance is not available, query forwarding can be random or based on the second and third items about neighbor CHs. The more files and more keywords in a neighbor cluster, the more queries forwarded to the CH of that cluster. The goodness GF of a neighbor CH c for forwarding the query q is estimated based on the past similar queries provided in its profile using the following formula. Only the top s similar queries, denoted by Q_s, are considered, where s is a system parameter. In the formula, q_i denotes a query in Q_s. $Sim(q, q_i)$ represents the similarity between q and q_i, and α is a system parameter used to control the influence of very similar queries.

$$GF(c, q) = \sum_{q_i \in Q_s} Sim(q, q_i)^\alpha$$

The similarity between queries are calculated using the well-known *cosine similarity model* [8]. Assume that the queries p and q are represented by the binary query vectors: $p = (p_1, p_2, ..., p_d)$, and $q = (q_1, q_2, ..., q_d)$. The angle between the two vectors are denoted by β. Then the similarity between p and q is computed using the following equation. Two queries are similar if their similarity score exceeds certain threshold.

$$\cos\beta = \frac{\sum_{i=1}^{d} p_i q_i}{\sqrt{\sum_{i=1}^{d} p_i}\sqrt{\sum_{i=1}^{d} q_i}}$$

4. Experimental results

The performance of our CIS searching is evaluated against the dominating-set-based (DS) searching through simulations. The query performance is in terms of the query

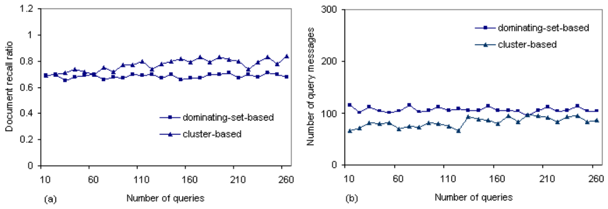


Figure 3. (a) The recall ratio of documents. (b) The number of query messages per query (average degree = 7).

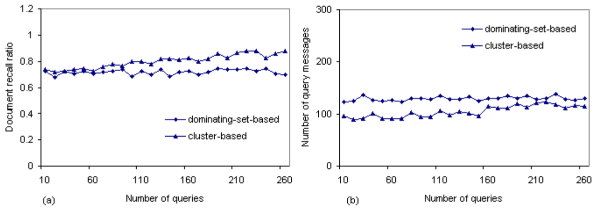


Figure 4. (a) The recall ratio of documents. (b) The number of query messages per query (average degree = 8).

efficiency and the quality of the query results. The query efficiency is measured by the number of query messages per query. The quality of the query results is represented by the document recall ratio, which is the number of documents in the query result divided by all desired documents in the system. We create a P2P network of 100 nodes. Two topologies are simulated: one regular random graph with average degree 7 and one denser random graph with average degree 8. 100 categories of files are randomly distributed among nodes such that each node stores average 150 files in a few categories. A file is indexed using the keywords for its category. Queries are generated from a set of keywords and arrive sequentially. The TTL value for both methods are 5 hops.

Figure 3(a) shows the recall ratio per query as the number of queries increases when the average node degree is 7. Clearly, the CIS searching always finds more documents than the DS searching because CIS forwards queries intelligently while DS blindly forward queries. It is also observed that the recall ratio of the CIS algorithm increases over time while that of the DS algorithm remains about the same. This is because a peer in CIS can learn from the past similar queries and intelligently forwards the query to peers that are more likely to have the desired documents. However, the DS algorithm does not learn. Figure 3(b) portrays the query efficiency per query when the average degree is 7. The CIS searching has a lower query cost than the DS searching. This is because the CIS searching utilizes a maximum independent set (MIS) and non-overlapping clusters. Similar results in Figure 4 are obtained when the average degree is 8. In summary, compared to the DS searching, the CIS searching offers a higher recall ratio and lower query cost as a result of its learning ability, intelligent query forwarding, non-overlapping clusters, and utilization of maximum independent sets.

5. Conclusions

In this paper, we proposed a cluster-based intelligent searching (CIS) algorithm in unstructured P2P networks. It first divides the entire P2P overlay into disjoint clusters each of which has a clusterhead (CH). CHs form a maximum independent set and are connected via gateways. A node that is neither a CH nor a gateway is an ordinary node. During the query processing, within each cluster, the CH performs the data lookup for the entire cluster using data summaries of non-CH members. Between clusters, CHs intelligently forward queries through gateways to the top neighbor CHs that are most promising in returning query results. Gateways do not conduct data look-up but only forward queries to the CHs specified in the received queries. Ordinary nodes are shielded from query traffic. Data summaries are stored in a compressed format using bloom filters. Intelligent query forwarding is based on the past similar queries answered via neighbor CHs. The CIS searching achieves a higher recall ratio at a lower query cost than the existing similar approaches. In the future, we plan to extend the basic CIS algorithm to intelligent searching that employs core-based clusters and k-hop clusters.

References

[1] X. Li, and J. Wu. Searching Techniques in Peer-to-Peer Networks. *Handbook of Theoretical and Algorithmic Aspects of Sensor, Ad Hoc Wireless, and Peer-to-Peer Networks, Edited by J. Wu*, CRC Press, 2005.

[2] C. Yang, and J. Wu. A dominating-set-based routing in peer-to-peer networks. *Proc. of GCC'03*, 2003.

[3] A. Sigla, and C. Rohrs. Ultrapeers: Another Step Towards Gnutella Scalability. *http://www.limewire.com*.

[4] A. Fisk. Gnutella Ultrapeer Query Routing v0.1. *http://www.limewire.com*.

[5] B. Bloom. Space time trade offs in hash coding with allowable errors. *Communications of ACM*, vol. 13, no. 7, 1970.

[6] J. Wu, and W. Lou. Forward-Node-Set-Based Broadcast in Clustered Mobile Ad Hoc Networks. *Wireless Networks and Mobile Computing, special issue on Algorithmic, Geometric, Graph, Combinatorial, and Vector*, vol. 3, no. 2, 2003.

[7] L. Fan, P. Cao, J. Almeida, and A. Broder. Summary Cache: A Scalable Wide-Area Web Cache Sharing Protocol. *IEEE Transactions on Networking*, vol. 8, no. 3, June 2000.

[8] M. Berry, and M. Browne. Understanding Search Engines: Mathematical Modeling and Text Retrieval. *Society for Industrial and Applied Mathematics (SIAM)*, 1999.

[9] V. Kalogeraki, D. Gunopulos, and D. Zeinalipour-yazti. A local search mechanism for peer-to-peer networks. *Proc. of ACM CIKM'02*, 2002.