Proceedings of the Federated Conference on
Computer Science and Information Systems pp. 257–261

ISBN 978-83-60810-51-4

Class-based Approach in Semantic P2P Information Retrieval

Ilya Rudomilov
Czech Technical University in Prague
Karlovo náměstí 13
Prague, Czech Republic
Email: rudomily@fel.cvut.cz

Ivan Jelínek
Czech Technical University in Prague
Karlovo náměstí 13
Prague, Czech Republic
Email: jelinek@fel.cvut.cz

***Abstract*—Peer-to-Peer (P2P) approach in information retrieval systems has drawn significant attention recently. P2P networks provide obvious advantages like scalability, reliability and, therefore, recent researchers are looking for a way to adapt these techniques to Information Retrieval fashion of nodes with heterogeneous documents. The greatest attention is paid to different semantic-based searching such as Gnutella Efficient Search (GES) proposed by Zhu Y et al., which derives from Vector Space Model. This paper proposes conceptual design of P2P unstructured information retrieval (IR) with heterogeneous documents on independent nodes.**

I. Introduction

SHIRKY defined *Peer-to-Peer (P2P)* in the article [22] in 2000 as:

> ...a class of applications that takes advantage of resources — storage, cycles, content, human presence — available at the edges of the Internet.

Peer-to-Peer networks have become popular for the success of applications like Napster [14], Gnutella [1], BitTorrent [6]. Although the beginning of the such networks was closely associated only with the file-sharing, modern projects pay attention to the possibility of use P2P scenario in Information retrieval background. P2P networks provide direct data exchange in overlay network because of distribution computational resources, content among large number of users. Each node in P2P fashion holds client and server responsibilities, which avoids a central server bottleneck and a single point of failure.

P2P networks can be classified according to level of decentralization (centralized, decentralized or hybrid) or to the control over data location and network topology (structured or unstructured). The general attention in recent publications is paid to unstructured (i.e. consist of nodes without global information, summarized indices of network, etc.) and decentralized (no any server, mediator) networks because of their full decentralization and easier maintaining topology without collecting global indices/descriptors and distributing among all nodes. However, this line of research is more costly due to query routing (no precise mapping of nodes). Our first paper [20] discusses about these classifications of P2P networks in details.

Noticeable, we should solve wide range problems with P2P network maintaining and query processing despite the many advantages of P2P. The first significant problem is continuously joining and leaving P2P networks by nodes, known as “churn“. System should have dynamic topology and be able to provide the same services regardless of the current network topology. The second is connected with first and is associated with query processing in dynamic network. The classical approach in unstructured P2P Gnutella-like networks is to use positive time-to-live (TTL) indicator to limit number of hops in a network: a query is transferred inside network until TTL expires.

II. Semantic Search in Unstructured P2P

Actually searching techniques in P2P networks would be classified as blind or informed. Informed search utilizes routing schemes to forward queries while blind has no information about other nodes and their content. Instances of blind search are random walk [10] or *k*-walker random walk [10]. Direct Breadth First Search [24] or routing indices [7] are informed. Moreover, searches in P2P can be classified as semantic or non-semantic. Semantic search is based on locating documents with similar semantic content (“Sport“, “Baseball“, “Cooking“), and non-semantic âĂ¿ on file IDs and does not need any semantic information.

Taxonomy of semantic searches depends of representation of documents [12], thus these networks can be classified as Ontology-based or IR-based. Ontology-based rely on ontology mappings of all document types on nodes. All used essences might be manually described or automatically generated by transformation already existing mappings. Extractor load data from documents under user-defined rules, thereby this type of systems try to apply Semantic Web ontology techniques to P2P networks. Ontology-based approach has high precision and performance; however it is expensive for systems with heterogeneous document collections. A range of IR-based semantic search networks like Gnutella Efficient Search [26], Class-based Semantic Searching Scheme [8] is more suitable for real networks because of heterogeneous occurrence of stored documents. It is hard to implement strict rules for user documents and subsequent document collection extraction; we have to parse documents without strict structure. IR-based search is the main trend in P2P IR for last years because of solving this problem by applying classical models

in Information Retrieval, such as Vector Space Model (VSM) [4] or Latent Semantic Indexing (LSI) [15].

III. Related work

A. Vector Space Model

The Vector Space Model (VSM) [4] is common technique in Information Retrieval. VSM provides model for representing text documents as frequency vectors of some identifiers like index terms. Each vector can be represented by matrix:

$$\begin{bmatrix} & T_1 & T_2 & \cdots & T_t \\ D_1 & w_{11} & w_{21} & \cdots & w_{t1} \\ D_2 & w_{12} & w_{22} & \cdots & w_{t2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ D_n & w_{1n} & w_{2n} & \cdots & w_{tn} \end{bmatrix} \quad (1)$$

Where *T* is of a set of stemmed with non-valuable stopwords ("is", "a", "the") terms, *D* represents a set of documents and *w* is a weight of respective term in respective document. Weight is represented by number of inclusions of terms in the document often.

Suppose a collection with 2 text documents, each of them has only one string: "The quick brown fox jumps over the lazy dog" and "He loves his dog" respectively. Table I shows representation of the collection.

As a consequence, we are able to use cosine measure [17] to calculate similarity of different documents. The cosine similarity is often used in information retrieval and range of evaluated projects.

B. Online spherical k-means clustering

There are different document clustering algorithms: *k*-means, fuzzy c-means, Gaussian mixture model, etc. We suppose that online spherical *k*-means (OSKM) [25] clustering is the most suitable for our project because of its high clustering performance, good results and relatively simple implementation. Algorithm is based on well-known Winner-take-all competitive learning [11], where each cluster centroid is incrementally updated given a document, system has not supervisor and is pretty suitable for decentralized solutions.

C. Gia

The Gnutella network is a popular, pure decentralized P2P solution for file sharing with flooding search. However, its nodes are extremely transient and problems with nodes availability were partially solved in non-semantic Gnutella-based search which called Gia [5]. Measurement by Microsoft researchers found that a median time of Napster or Gnutella nodes uptime is about just 60 minutes [21]. Imagine situation in network with 100.000 nodes, this implies a churn rate of over 1600 nodes coming and going per minute. Updating global information on all nodes is too expensive, using centralized servers is out of decentralization policy and therefore authors of Gia proposed new topology adaptation algorithm.

Each Gia node contains a host cache with a cached list of other Gia nodes (their IP address, port number, and capacity). The adaptation algorithm is based on independent calculation a level of satisfaction (S) of this list, i.e. how current node is satisfied with its neighbours (1 is absolutely satisfied and 0 is for quite unsatisfied, respectively). As long as node is not satisfied topology adaptation algorithm tries to find new neighbours for it by random selecting other node and compare its characteristics under a level of satisfaction metrics. This handshake method is principally new in Gnutella-like networks and may be used in our project.

D. Gnutella Efficient Search

We can find a first attempt of using at least some above technologies in IR context in GES project (Gnutella Efficient Search) [26] by Zhu et al. The network is implemented in Semantic-based IR manner: periodically issued node summarized vectors are randomly sent to other nodes for semantic comparison.

Like to Gia, GES uses handshake protocol to compare characteristics of nodes (node vectors, in this case) and establish semantic links between nodes with documents with same semantic context (and further searching inside semantic cluster) or random links to nodes from definitely another cluster (and forwarding non-relevant queries). Replication of node vectors is selective, only one-hope and used only for semantically non-relevant node vectors (i.e. from node with random link).

The search protocol is clear and uses biased walk in semantically non-relevant cases and flooding in case semantically relevant queries. Received query is utilized by local search component, which tries to find relevant documents on the node. If it found at least one document, the node is called a semantic group target node and node uses semantic links to flood other nodes from current cluster.

E. Class-based Semantic Search Scheme

Our project is relative close to such Gnutella-like IR system CSS (Class-based Semantic Search Scheme) [8] by Huang et al. CSS is founded on main ideas of GES project, however the main invention is in possibility of splitting node vector into different class vectors and clustering them by OSKM. This has obvious reasons, because in real systems nodes have heterogeneous document collections with different topics. Possible, documents about sport, cooking and programming would be stored on the same node and it would be unobvious to create summarized node vector for them. Moreover, CSS provides a novel formula to calculate the relevance between class vectors.

IV. Our approach

A. Node structure

Preliminary structure of nodes was generally borrowed from [9] and discussed in our last paper [20]. Now we had to update and extend it according to semantic approach for P2P systems.

TABLE I
VSM REPRESENTATION OF 2 DOCUMENTS

Term	brown	dog	fox	jumps	he	his	lazy	loves	over
Document 1	1	1	1	1	0	0	1	0	1
Document 2	0	1	0	0	1	1	0	1	0

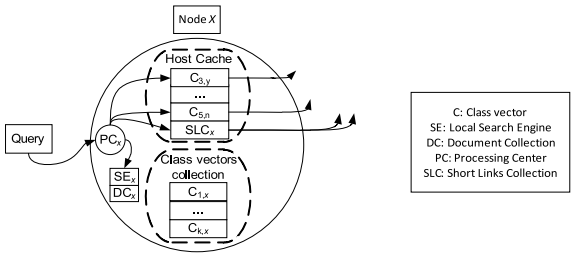

Fig. 1. Structure of node

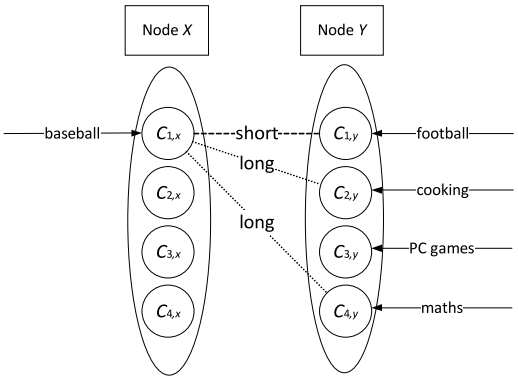

Fig. 2. Two nodes with four class vectors

Now, each node consists of the following main components (Figure 1).

- A set of documents (*Document collection* or *DC*) with available heterogeneous documents. Document Collection is used by *Local Search Engine (SE)* in searching progress and generating *Class vectors (C)*.
- Generated by VSM and semantically split with OSKM a set of *Class vectors (C)*. Class vector contains summarized vector of all semantically relative documents from Document collection on current node. It is a key component, which is distributing through the network to find semantically close Class vectors and add them into a list of neighbours. Undoubtedly, each Class vector has information about node (e.g. IP-address).
- *Processing centre (PC)* for incoming queries, returning results of searching and forwarding a query to neighbour nodes according to Search protocol. All requests in the network are passing throw nodes Processing Centres.
- *Host cache (HC)* contains information about neighbour nodes.
  - *Long links collection (LLC)* with corresponding Class vectors from several definitely semantically non-related nodes. This collection is used for forwarding semantically non-related queries.
  - *Short links collection (SLC)* with information (e.g. IP-addresses) about nodes with semantically related Class vectors to current local Class vector.

B. Class vectors

One of the most important innovations in IR is using several class vectors instead of summarized node vector. This concept is proposed by several projects and has proven efficiency in CSS simulation [8]. Each node has at least one Class vector with a summarized vector of semantically related local documents. These Class vectors are calculated by Vector Space Model, as follows. We are using a typical for class-based networks model of calculating from vector of every document to summarized node vector and split them by OSKM into a given the number of classes we want to cluster (e.g. 5). The same weight algorithm is used in CSS:

1) A term's weight is assigned to its frequency in a document.
2) In "dampened" vector all weights are recalculated by *tf* scheme in the form:

$$1 + log(tf) \tag{2}$$

3) A weighed term vector is normalized to unit length.
4) All generated term vectors (or node vector) are processed by OSKM [25] algorithm for splitting into required number of class vectors.

C. Semantic links

Semantic approach in our project consists in semantic links among semantically distributed related class vectors, which create a resulting semantic cluster. There are two types of semantic links:

- *Long link* to connect with semantically non-related class vector. For this type connection storing a replica of class vector is required.
- *Short link* to connect semantically related class vector. Storing of class vector replica is not required.

Decision of creating a link is based on relevance score between two respective class vectors. In connection with applying VSM techniques we are using cosine similarity of class vectors. The short and long links can be considered as semantic and random links in GES.

Figure 2 illustrates instance of two nodes with four class vectors. Class vector "Baseball" on left node has a short link

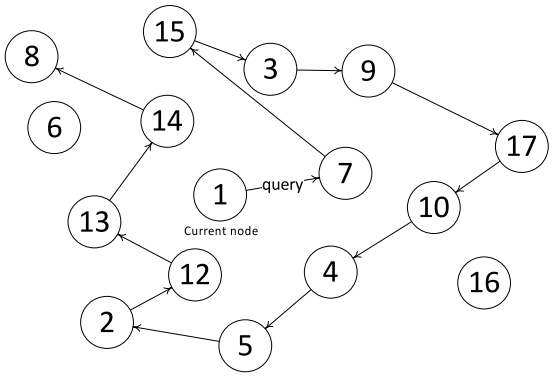

Fig. 3. A routing of topology query message

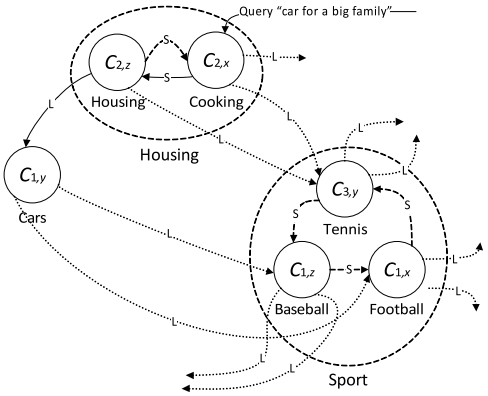

Fig. 4. Example of searching into semantic cluster

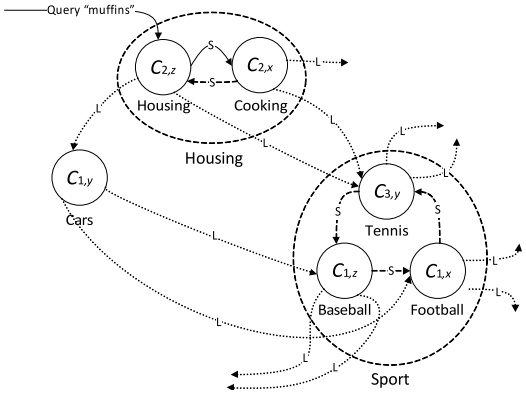

Fig. 5. Example of searching in another semantic cluster

with right node "Football" because of similar semantic content of respective document collections. On the other side, long links are created to "Cooking" and "Math", because these topics are not similar to sport.

D. Topology Adaptation

Used topology adaptation mechanism is similar to evaluated in CSS and aims to find the most relevant semantic neighbours to create short links in one hand, and the most non-relevant for long links (and forward a non-relevant queries).

When a node joins a network, it first tries randomly connecting (Figure 3) to other nodes using typical Gnutella bootstrap mechanism.

It periodically send random walk *topology query message* to find other interested-in nodes, it is possible because of encapsulating all class vectors into a query. TTL limitation and defining a maximum number of answers is used to avoid situation of overloading by topology query. Received answers (in the same shape as topology query) are comparing with local nodes and the most suitable candidates will be saved in host cache.

Host cache is updated continuously. Node continues to receive topology queries from new nodes or looking for new neighbours, compare them and collecting suitable candidates into host cache. Similarly, node starts to send topology query when some neighbour leaves a network. Failed nodes would be detected by keep-alive messages from their neighbours.

E. Search protocol

The proposed search protocol is content and class-based. Query is processed in two modes: in *direct walk mode* to find a relevant semantic cluster through long links (because of their class vectors) and then *flooding mode* within cluster through short links (because of their involving into same relevant semantic cluster).

Figure 4 illustrates example of searching into semantic cluster (class vectors are shown). Node Z received query "muffins" and forwarded it to neighbour node X through short link (they are in the same cluster).

Figure 5 illustrates another example, searching in another semantic cluster (class vectors are shown). Node X received query "car for a big family" and firstly forwarded it to neighbour node Z through short link (query is similar to their context at least a little bit) and then to node Y through long link.

V. Conclusion

In this paper, we propose combination of CSS and GES models as Gnutella-like Information Retrieval system. As the Gnutella, our P2P network is unstructured with a view to scaling on the one hand, however borrowed from Gia topology query distribution and handshaking protocol prevent blind flooding over whole network.

Node is consist of several components: Document Collection, Local Search Engine, Processing Centre, summarized Short and Long links into Host Cache and source, Document Collection. Document Collection is processed throught Vector Space Model and Online Spherical k-means Clustring to generate several semantic class vectors as identifiers of node.

VI. Future work

Our system design has reason to be productive and effective as is a combination of the various productive and proven

techniques. In the near future we plan to conduct with PeerSim [13] series of evaluations to compare the performance and scalability in comparison with the structured P2P architectures such as CAN [18], Pastry [19], and Chord [23]. It makes sense to check and efficiency compared to traditional Gnutella, despite a number of existing publications on this subject.

We plan to implement in the long term this project using the open-source framework JADE (Java Agent Development framework) [3], what is a free framework for developing Java-based intelligent multi-agent systems and in addition, according to standards from the FIPA (Foundation for Intelligent Physical Agents) [2], a major non-commercial group in the multi-intelligent systems. The FIPA's membership includes Toshiba Corp., Siemens, Boeing Company, RWTH Aachen University, etc. The widely adopted FIPA standards are the Agent Management and Agent Communication Language (FIPA-ACL) specifications, which already in use as an industry standard. It is a modern and popular environment, which can be used without restriction or need major interventions and other collaborators in the research. One can certainly believe that the principles of the proposed system will be used not only as a research subject, but in practical applications. Using JADE for implementation MAS-based P2P applications is a common practice [16] and has obvious advantages:

- Interoperability: JADE is according to FIPA specifications;
- Portability: Java allows to use different platforms and JADE-based implementation can run on J2EE, J2SE, J2ME environment;
- Easy of use: JADE is a set of APIs, which has GUI for a nodes management.

REFERENCES

[1] The Gnutella Protocol Specification V0.4. http://www.stanford.edu/class/cs244b/gnutella_protocol_0.4.pdf.

[2] FIPA specifications, 2000. http://www.fipa.org.

[3] JADE software framework, 2011. http://jade.tilab.com.

[4] M. W Berry, Z. Drmac, and E. R Jessup. Matrices, vector spaces, and information retrieval. *SIAM review*, pages 335–362, 1999.

[5] Y. Chawathe, S. Ratnasamy, L. Breslau, N. Lanham, and S. Shenker. Making gnutella-like p2p systems scalable. In *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 407–418, 2003.

[6] B. Cohen. The BitTorrent protocol specification, 2008. http://www.bittorrent.org/beps/bep_0003.html.

[7] A. Crespo and H. Garcia-Molina. Routing indices for peer-to-peer systems. In *Distributed Computing Systems, 2002. Proceedings. 22nd International Conference on*, pages 23–32, 2002.

[8] Jun-Cheng Huang, Xiu-Qi Li, and Jie Wu. A semantic searching scheme in heterogeneous unstructured P2P networks. *Journal of Computer Science and Technology*, 26(6):925–941, November 2011.

[9] H. Z.V Lesser. *Toward Peer-to-Peer Based Semantic Search Engines: An Organizational Approach*. VDM Verlag, 2008.

[10] Q. Lv, P. Cao, E. Cohen, K. Li, and S. Shenker. Search and replication in unstructured peer-to-peer networks. In *Proceedings of the 16th international conference on Supercomputing*, pages 84–95, 2002.

[11] Wolfgang Maass. On the computational power of Winner-Take-All. *Neural Comput.*, 12(11):2519–2535, November 2000.

[12] A.R. Mawlood-Yunis, M. Weiss, and N. Santoro. From p2p to reliable semantic p2p systems. *Peer-to-peer networking and applications*, 3(4):363–381, 2010.

[13] A. Montresor and M. Jelasity. PeerSim: a scalable P2P simulator. In *Peer-to-Peer Computing, 2009. P2P '09. IEEE Ninth International Conference on*, pages 99 –100, September 2009.

[14] L. L. C. Napster. Napster, 2001. http://www.napster.com.

[15] C. H Papadimitriou, H. Tamaki, P. Raghavan, and S. Vempala. Latent semantic indexing: A probabilistic analysis. In *Proceedings of the seventeenth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems*, pages 159–168, 1998.

[16] A. Poggi and M. Tomaiuolo. Integrating peer-to-peer and multi-agent technologies for the realization of content sharing applications. *Information Retrieval and Mining in Distributed Environments*, pages 93–107, 2011.

[17] J. Pokorny. Web searching and information retrieval. *Computing in Science Engineering*, 6(4):43 – 48, August 2004.

[18] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker. A scalable content-addressable network. *ACM SIGCOMM Computer Communication Review*, 31(4):161–172, 2001.

[19] A. Rowstron and P. Druschel. Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems. In *Middleware 2001*, pages 329–350, 2001.

[20] I. Rudomilov and I. Jelinek. Semantic p2p search engine. In *2011 Federated Conference on Computer Science and Information Systems (FedCSIS)*, pages 991–995. IEEE, 2011.

[21] S. Saroiu, P. K Gummadi, S. D Gribble, et al. A measurement study of peer-to-peer file sharing systems. In *proceedings of Multimedia Computing and Networking*, volume 2002, page 152, 2002.

[22] C. Shirky. What is p2p... and what isn't. *OpenP2P.com*, 2000. http://openp2p.com/pub/a/p2p/2000/11/24/shirky1-whatisp2p.html.

[23] I. Stoica, R. Morris, D. Karger, M. F Kaashoek, and H. Balakrishnan. Chord: A scalable peer-to-peer lookup service for internet applications. *ACM SIGCOMM Computer Communication Review*, 31(4):149–160, 2001.

[24] B. Yang and H. Garcia-Molina. Improving search in peer-to-peer networks. In *Distributed Computing Systems, 2002. Proceedings. 22nd International Conference on*, pages 5–14, 2002.

[25] Shi Zhong. Efficient online spherical k-means clustering. In *Neural Networks, 2005. IJCNN '05. Proceedings. 2005 IEEE International Joint Conference on*, volume 5, pages 3180 – 3185 vol. 5, August 2005.

[26] Y. Zhu, X. Yang, and Y. Hu. Making search efficient on gnutella-like p2p systems. In *Parallel and Distributed Processing Symposium, 2005. Proceedings. 19th IEEE International*, page 56a, 2005.