HiWaRPP — Hierarchical Wavelet-based Retrieval on Peer-to-Peer Network

Mihai Lupu Bei Yu
Singapore-MIT Alliance
National University of Singapore
E-mail: mihailup,yubei@comp.nus.edu.sg

Abstract

This paper introduces the use of wavelets for information retrieval in a peer-to-peer environment. In order to achieve our purposes, we use a new combination between broadcasting and a hierarchical overlay. Compared to previous approaches, we do not store complete information about the children of a super-peer, nor do we broadcast the queries blindly. We approximate the feature vectors using the multiresolution analysis and the discrete wavelet transform. Each peer is represented by a high-dimensional feature vector and the height of the hierarchy is logarithmic in the dimensionality of this feature vector. Leaf nodes represent real peers, while internal nodes are virtual peers used for routing. Our retrieval method has been tested with both real and synthetic data and shown to be efficient in retrieving relevant information, resulting in good precision and recall on four standard test collections.

1 Introduction

Semantic-based retrieval is essential for data sharing applications in peer-to-peer networks. This is why we are willing sometimes to trade local storage space for higher retrieval efficiency and effectiveness. However, resources are still limited and maintaining global information of the order of the entire network is impractical.

In general, existing work in the peer-to-peer realm has been divided between the commercial applications that use mostly unstructured or centralized models [3, 1] and research-oriented structured networks [5, 8]. Super peer methods introduce a set of nodes that collectively take over the role of the central server. Such methods have been frequently studied [2, 4, 7] with good results.

Our work provides a flexible retrieval framework that is scalable in terms of routing information stored in individual peers. Each peer provides a feature vector (ID) that summarizes its content to be shared in the P2P network. We approximate the feature vector with the multiresolution analysis (MRA) and the discrete wavelet analysis (DWT). Our overlay network is a hierarchical structure that consists of both virtual peers and real peers, where real peers are the leaf nodes in the hierarchy. The top levels of the hierarchy are superimposed with a fully connected graph, such that query load is distributed equally among the top nodes. Different levels of approximation generated by MRA are distributed to the different levels of the network hierarchy, with the highest level containing the coarsest information. We route the query towards the peers that contain similar content with routing performance comparable to DHT-based methods. However, our framework does not need to change peer IDs upon insertion in the network, thus maintaining the semantic information associated with them.

Our main contributions are summarized as follows:

- an intuitive network overlay, without the overhead of moving documents across peers
- an effective information retrieval method to route queries towards peers that hold relevant data
- reduced storage utilization by using approximations of the real peer IDs

2 HiWaRPP system prototype

In our prototype system, each peer shares a document collection, and it provides a summary of it as the feature vector. The approximation of IDs is propagated up in the hierarchy and the nodes at higher level store more, but coarser approximations for their descendants.

When a node joins the network, it locates the lowest-level node with an incomplete number of children and,

if necessary, creates and stores a series of virtual peers to insert itself.

2.1 Routing table

Each virtual peer in the network maintains a list of its own children and a link to its parent. For each of its children, it stores which real peers are accessible through that child. With this simple technique, the HiWaRPP framework is very flexible in the sense that

1. if a peer changes slightly its ID due to the insertion of the new document, the higher level approximations (the coarser ones) remain mostly unchanged. Thus, a search before the update is actually performed, will still find the correct peer.
2. as the query travels down the network, it receives more information about its destination and can make decisions as whether to continue or not.

Route selection. When a real peer has a query, it forwards it to its parent and higher up the hierarchy as much as possible. At higher levels, the virtual nodes have a broader view of the network and are able to select potentially relevant peers. After the choice has been made, the query is sent downwards back to the leaf level where the real peers are.

2.2 Bandwidth and storage requirements

The super-peers of the overlay are subjected to stress due to a high number of messages they have to receive and, in some cases, reply to. For both types of messages (update and query), their size is not an issue, as they are reduced with the same method that has been used to reduce the amount of storage needed. The problem lies mainly in opening and closing a large number of connections to this super-peer.

Fortunately, due to the approximation used, update messages do not reach the top of the hierarchy every time a new peer joins or a new document is inserted by an existing peer. Updates are propagated towards the root only when their added information surpasses a certain threshold. Periodical corrections are scheduled to correct updates error accumulated over time.

On the other hand, query messages should always reach the root in order to get a global view of the network. This is why, at the top levels, we superimpose a fully connected graph. We practically transform the top hierarchy in a cluster of super-nodes, with the difference that we still maintain the hierarchy in order to manage the data distribution in the cluster. The root of the hierarchy no longer answers queries directly, but is used solely to gather data and distribute it to

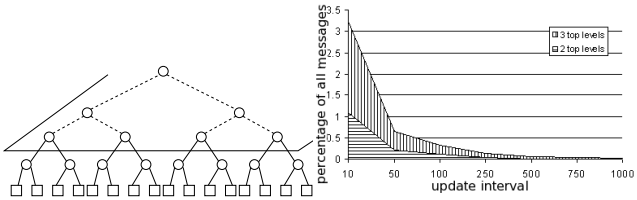


Figure 1: a. System overview b.Proportion of update messages

the lower-level virtual peers that will effectively answer queries (Figure 1a).

The number of levels that participate in this knowledge sharing depends on the size of the network. Figure 1b shows the proportion of update messages as a function of the number of levels and the frequency of updates.

Using multiresolution, we show that we can store information about all the peers without overwhelmingly increasing the memory requirements at each peer. Figure 2a shows the worst case scenario for storage consumption. Given the same expected number of peers in the network, smaller buckets impose more storage due to the necessity of a higher tree. The slope tends to decrease due to the fact that as more peers join the network, their approximated IDs are added higher in the hierarchy and, as we get closer to the root, the size of the approximations decrease exponentially while the number of virtual peers increases only linearly.

3 Experimental study

Efficiency of the retrieval. We have implemented a simulator and tested it with as high as 1 million 'real' peers.

First we look at the maintenance messages by analyzing the amount of traffic generated when inserting up to 100k nodes. Figure 2b shows that the average number of messages grows sublinearly in the number of nodes. The reason for this behavior is the way virtual peers are distributed on the real peers. As the network size increases, there is relatively more communication locally - between the virtual peers that are stored on the same real node.

Figure 2c. shows how the average size of the messages decreases logarithmically with respect to the number of peers. This is due to the fact that in a larger network, the levels of approximations decrease logarithmically and, consequently, so do the average sizes of the messages. It also shows that increasing the bucket size increases the average size of a message. This is because larger buckets make the network more flat, and relatively more messages are circulated at the lower leves, where the approximations are larger.

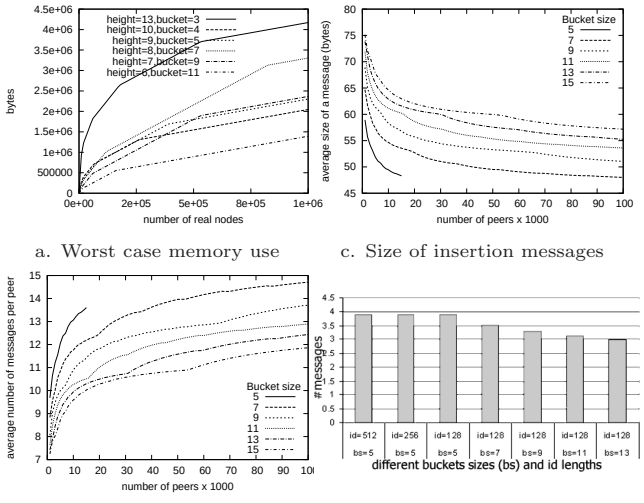
Figure 2d. shows the average number of messages per query for varying bucket sizes and ID lengths. We observe that the number of messages is less than logarithmic in the number of nodes in the network.

Effectiveness of the retrieval. We evaluated the effectiveness of the retrieval with four benchmark collections of documents. In order to test with larger number of peers, we divided the four collections into 30 peers by topic. Each peer results in slightly more than 200 documents, which are from one of the collections. Each peer generates its summary for its own collection, which is a vector of document frequencies for the terms in the collection.

We compare our retrieval precision and recall with the standard vector space retrieval model (VSM). Since euclidean distance is applied in our approximation space, we also use euclidean distance to compare query vector and document vectors in VSM. Figure 3 presents the comparison results for two of the four query sets associated with each collection.

From the figures we observe that our retrieval method outperforms VSM. The results actually surpassed our expectations, as we estimated that it should behave just as good as the naive VSM method. Our assumption as to the cause of these better results lies mainly in the way the vector IDs are created. When the collections are parsed and inserted into the network, they are done sequentially. That results in more or less compact subsets of the vectors with non-zero values. Common terms among collections are scattered, but generally surrounded by zero values. The approximation maintains the non-zero regions, while fading out the common words.

The observed improvement demonstrates that our method is promising and its potential usefulness for other applications that prefer the euclidean distance measure, such as video retrieval [6].



a. Worst case memory use c. Size of insertion messages

b. Average number of messages d. Messages per query

Figure 2: Efficiency measurements

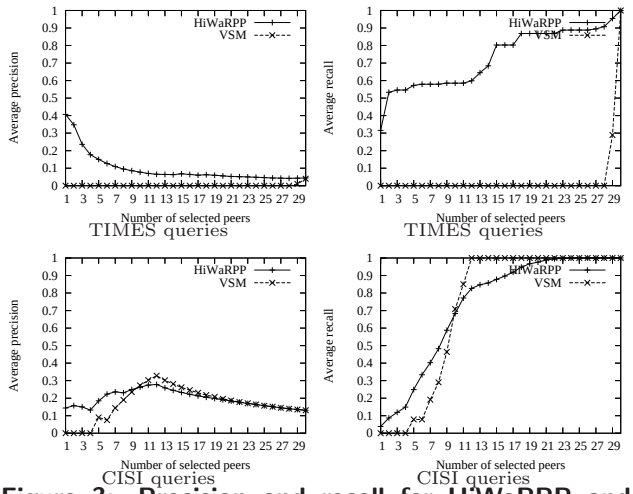


Figure 3: Precision and recall for HiWaRPP and VSM.

4 Conclusions

In this work we investigate the usage of wavelet decomposition for a hierarchical representation of the routing table in a peer-to-peer environment. The new approach behaves at least as good as VSM retrieval method but shows great flexibility in balancing the speed and accuracy of the retrieval, as well as distributing information in a peer-to-peer network in an easy to understand and maintain fashion.

Our prototype shows promising results, is based on a solid mathematical foundation and it has been tested with both real world and simulated documents.

References

[1] Napster web page. Web Document. www.napster.com.

[2] S. Daswani and A. Fisk. Gnutella udp extension for scalable searches v0.1. Web Document, 2002. http://cvs.limewire.org/fisheye/viewrep/~raw,r=1.2/limecvs/core/guess_01.html.

[3] Limewire. Gnutella protocol v0.4. Web document, 2004. www9.limewire.com/developer/gnutella_protocol_0.4.pdf.

[4] W. S. Ng, B. C. Ooi, and K. L. Tan. BestPeer: A self-configurable peer-to-peer system. In *Proc. of the* 18^{th} *ICDE*, 2002. Poster Paper.

[5] S. Ratnasamy, M. Handley, R. Karp, and S. Shenker. Application-level multicast using content-addressable networks. *LNCS*, 2233, 2001.

[6] H. T. Shen, B. C. Ooi, and X. Zhou. Towards effective indexing for very large video sequence database. In *Proc. of the* 24^{th} *ACM SIGMOD*, 2005.

[7] H. T. Shen, Y. F. Shu, and B. Yu. Efficient Semantic-based Content Search in P2P Network. *IEEE Transactions on Knowledge and Data Engineering*, 16(7):813–826, 2004.

[8] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan. Chord: A scalable peer-to-peer lookup service for internet applications. In *Proc. of the Conf. on Applications, Technologies, Architectures, and Protocols for Computer Communications*, 2001.