Towards a Superimposed Peer Infrastructure for iNformation Access *

[EXTENDED ABSTRACT]

Emanuele Di Buccio, Nicola Ferro and Massimo Melucci

Department of Information Engineering, University of Padua, Italy
{dibuccio, ferro, melo}@dei.unipd.it

Abstract. The Peer-To-Peer (P2P) paradigm is a promising approach for several distributed applications, among which distributed storage systems. SPINA is a software architecture that aims at encompassing indexing and retrieval of unstructured documents stored in a P2P network. This paper describes the current status of the design and the implementation of this software architecture.

1 Introduction

The advent of P2P networks on the scene of the search engines poses new challenges for Distributed Information Retrieval (IR) and Digital Library (DL) Systems. The basic rationale of the P2P paradigm is that entities constituting the network are peers which can function as both client and server. P2P networks are a suitable solution to provide federated search capability to a large number of collections on the Internet and DL's, in an effective, convenient and cost-efficient way that is decentralized in nature [10].

Since a peer can join the network by connecting to any peer, a peer might be reached through intermediate peers thus requiring resource selection and query routing algorithms. Indeed, a peer can be connected to more than one peer and therefore a decision concerning the peer to which the query should be routed has to be made. Resource selection in P2P systems is therefore related to the task of query routing because of the topology of the network.

A network can be either structured or unstructured. The former kind of networks are based on a predefined structure — Distributed Hash Table (DHT) are often implemented in structured networks as they provide shared and efficient storage and access to keys, that is, the descriptors of the documents stored in the peers. Structured P2P networks enable efficient query routing, but requires an high degree of collaboration between peers. In *unstructured* networks, there is not any global data structure which stores the information about the content of the documents of the network. Hybrid networks are unstructured networks where

* This extended abstract is partially based on the paper *Content-based Information Retrieval in SPINA*, accepted to be published in the Proceedings of the 4th Italian Research Conference on Digital Library Systems (IRCDL 2008).

some peers, called ultra-peers, with previously established attributes automatically take over the central indexing server functions. Each ultra-peer is elected from normal peers and each one serves a group of normal peers. If each peer refers to one and only one ultra-peer, the networks is called *hierarchical*. Ultra-peers communicate to form the backbone of hybrid decentralized networks. The hierarchical P2P networks are more complex and have higher communication costs than those structured, yet they may provide best-match search techniques and require lower degree of collaboration than structured networks. The presence of the ultra-peers "enables directory services to automatically discover the contents of (possibly uncooperative) collections, which is well-matched to networks that are dynamic, heterogeneous, or protective of intellectual property" [10].

When P2P networks federate IR systems, the lack of information due to the limited knowledge peers have about each other causes a loss of recall. In order to reduce that loss, the network has to be explored as much as possible, in a way search efficiency is not limited by the high communication costs. A possible solution to the problem of loss of recall is to select, and to route the queries to the resources (e.g. the peers) which most probably store information relevant to the user's information need.

It is our opinion that the design of a P2P-IR system should be done both at a modelling and at an architectural level. A weighting scheme was proposed in [7] for addressing the design of a P2P-IR system at modelling level. In [5] the problem was addressed at an architectural level by introducing a software architecture called Superimposed Peer Infrastructure for iNformation Access (SPINA). In the first part of this paper the designed architecture and the weighing scheme will be briefly reviewed, in order to make clear the description, reported in Section 3, of some of the implementation choices made. Section 3 will illustrate also some issues emerged during the development of SPINA, which are the aspects we are investigating at the present time.

2 SPINA

The main characteristics of SPINA can be summarized as follows [3]:

1. it aims at being independent of both the underlying network infrastructure and the media of the documents stored in the network;
2. it is focused on exchanging statistics about the features extracted from the full-content of indexed documents and aggregating the resources according to the level hierarchy;
3. it selects peers and routes query by the probability that a peer or a document store relevant information — this approach does not require any clustering.

Each of these aspects will be deepened in the following subsections.

2.1 The Architecture

SPINA aims at being independent of the underlying network infrastructure: different network topologies are supported, ranging from unstructured, to hybrid and hierarchical. In terms of the architecture, Figure 1 depicts two important features of SPINA. The first is that the proposed software architecture "superimposes" different logical layers over an existing infrastructure.

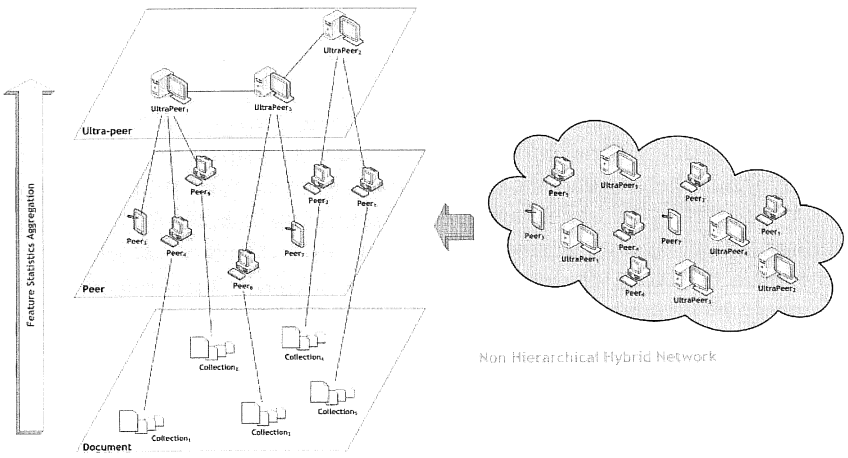

Fig. 1. SPINA Layers.

The instance depicted in the figure, which reflects the current implementation of SPINA, is structured on three levels: starting from the "lower" one, we can enumerate the (1) document, the (2) peer and the (3) ultra-peer level. The way SPINA was designed and the weighting scheme adopted — see Section 2.3 — allow the approach to be generalized for an arbitrary number of levels. The other feature depicted in Figure 1 is that the considered architecture is not only layered, but also hierarchical: indeed each peer refers to one and only one ultra-peer. This hierarchy of peers together with the adopted resource selection strategy allow the number of messages to be reduced during the query routing.

2.2 Statistics Exchange and Search

In the considered architecture, each peer is provided with a local search engine to which the user submits its queries. The local search engines perform all the indexing and retrieval operations. What is needed for making retrieval across peers possible is performed by SPINA. Each query is formulated as a bag of features. Features can be of different types depending on the different media that characterize a document; for example keywords or music patterns are features, respectively, of textual or audio fragment of a document.

If the end user requests for a P2P search when interacting with a peer, the query is routed to the ultra-peer to which the peer refers. Each ultra-peer manages a local index which stores summary information about the content of the peers in the group that serves. Basically, an ultra-peer associates the list of peers to each feature, as well as the total weight of every feature occurring in every peer – of course, no data is stored about the peers which do not store a feature. These indexes are obtained by exchanging statistics about the features extracted from the indexed documents and aggregating the resources according to the level hierarchy – as illustrated by the arrow depicted on the left of the SPINA layers in fig. 1. According to this information an ultra-peer selects peers and routes query by the probability that a peer or a document stores relevant information. Peers return to the referring ultra-peer a ranked list of objects in answer to the formulated query.

The P2P search is not only restricted to the group the starting peer belongs to. In fact, ultra-peers communicate each other to form the backbone of hybrid decentralized networks. Each ultra-peer maintains an index which stores information about its neighbours, obtained by the exchange of statistics previously mentioned. Similarly to the index about the peers of a group, an ultra-peer associates the list of neighbouring ultra-peers to each feature, as well as the total weight of every feature occurring in every neighbouring ultra-peer; no data is stored about the neighbouring ultra-peers which do not store a feature. Respect to the aggregation process depicted in fig. 1, the last level requires an "horizontal" aggregation: in fact the aggregation of information concerns its peers – same hierarchy layer – and not resources at lower layers. This aggregation of information allows for selecting ultra-peers to which the query is forwarded.

2.3 The Weighing Framework

Since IR is intrinsically affected by uncertainty, and the latter increases when the information seeking activity is performed across a P2P network, the use of algorithms which rank resources at the different levels based on probabilistic models is a natural modelling option [9]. These algorithms allow the peers of a group to be ranked by the probability that the documents store relevant information. Similarly, the neighbouring ultra-peers will be ranked by the probability that the documents of the peers of their groups store relevant information. The top ranked k peers or ultra-peers will be selected. The resource selection problem in the considered network topology is characterized by a recursive nature exploited in the design of SPINA [5], but even before in the weighing framework proposed in [7]. The ranking scheme proposed looks like a TF·IDF scheme, but its components are in turn Term Weighted Frequency (TWF)·Inverse Resource Frequency (IRF) schemes which are recursively defined on top of hierarchy of types of peer. The IRF is a generalization of the IDF for the higher resource levels. This generalization was adopted in Distributed IR [6] and P2P-IR [8], respectively to rank collections and peers. On the contrary, TWF is peculiar of this scheme. For each level z, the TWF·IRF is defined as follows:

$$w_{i,j,t}^{(z)} = twf_{i,j}^{(z)} \cdot irf_{i,t}^{(z)}, \tag{1}$$

where i refers to the feature i, j refers to $r_j^{(z)}$, i.e. the resource j at level z to which the feature belongs, and t refers to $r_t^{(z+1)}$, which is the resource of level $z+1$ to which $r_j^{(z)}$ belongs. The TWF of a feature i w.r.t. a resource $r_j^{(z)}$ is computed as

$$twf_{i,j}^{(z)} = \left(\sum_{s=1}^{N_j^{(z-1)}} twf_{i,s}^{(z-1)} \right) \cdot irf_{i,j}^{(z-1)}, \tag{2}$$

where $N_j^{(z-1)}$ is the number of resources of level $z-1$ in $r_j^{(z)}$, $twf_{i,s}^{(z-1)}$ is the TWF of the feature i in $r_s^{(z-1)}$. The IRF can be computed by generalizing the IDF for the higher levels. The score of a resource $r_j^{(z)}$ w.r.t. a formulated query q is computed as $\sum_{i \in q} w_{i,j,t}^{(z)}$. Equation 1 can be applied at each level $z \geq 1$, while Eq. 2 for $z > 1$ — here documents are not considered as structured by sub-resources. Therefore the weighing framework, as the SPINA software architecture, "supports" an arbitrary number of levels. A characteristic of this framework is that resources including features which occur within few resources are top ranked, so the framework supports selecting few resources by thus helping minimize bandwidth.

An instance of this weighing framework was proposed in [11]. At the document level, the TF·IDF scheme was adopted. Then Eq. 1 and Eq. 2 were used to compute the weight of a feature in a peer. The weight of a feature in an ultrapeer was computed by considering only the contribution of the TWF of such feature in its peers. In [11] the efficacy of the ranking scheme is investigated: the first peer or ultra-peer visited gives the largest proportion of recall, thus confirming the hypothesis. One of the benefits of this weighing framework is that only little information aggregated according to the level hierarchy is required to rank peers and ultra-peers, thus helping the reduction of the network load.

3 SPINA Development

This section describes some of the choices made to implement the SPINA software architecture and points out some issues due to the topology of the network and the weighing model adopted, which we are going to tackle.

3.1 Implementing SPINA

At the present time SPINA adopts the JXTA technology to implement the communications among peers, particularly the Java implementation JXTA JXSE 2.5 [1]. JXTA provides the functionalities to implement an hybrid hierarchical P2P networks, that is the type of network supported by SPINA.

In order to implement the indexing functionalities required to store the information needed by the weighing scheme, the Apache Lucene library [2] was adopted. It was decided to use Lucene because it provides efficient implementation of the basic IR functions, the latter being an important requirement for

P2P-IR where peers can deliver little computational power to the network. As regards the local search, an interface between Lucene and SPINA was implemented in order to preserve the independence of the latter from the specific search engine library adopted. At the document level the functionalities the Lucene library provides to indexing and search documents may be directly used. But in P2P-IR not only documents have to be indexed and retrieved, but also peers and ultra-peers, which are provided with computational capabilities other than informative content like documents. For this reason, at the peer and ultra-peer levels Lucene was adopted only to implement the indexing functionalities, that is to store the information required by the adopted weighing framework and to efficiently access to them. Let consider for instance the peer-granularity index, that is the index where an ultra-peer stores the information needed to rank peers in its group. The peer-granularity index stores for each feature the list of peers where the feature is present and the weight associated to considered feature in the specific peer. The weights computed by the TWF·IRF scheme are real numbers. In order to store these values, a recent functionality of Lucene was used, that is the possibility to associate a metadata with each term position in a specific document. The basic rationale was mapping a peer to a Lucene Document, that is a record in the index, and a feature to a Lucene Term, that is the Lucene's unit of indexing – e.g. a word. The weight of the feature in the peer was stored as Payload, that is the metadata, of the Term in the Document. In order to rank peers according to this information, a ranking algorithm based on Document-At-A-Time (DAAT) strategies was implemented – DAAT algorithms guarantee a smaller run-time memory footprint. The same "mapping" was adopted to implement the indexing functionalities at the ultra-peer level.

3.2 Definition of the Neighbours

After describing the architecture and the weighing scheme, the notion of neighbour of an ultra-peer is introduced in this section. Each ultra-peer stores the information about its neighbours into its ultra-peer granularity index. However, the neighbours to be contacted for routing the queries can not be the neighbours suggested by JXTA. For instance JXTA, or in general a communication layer, can make every ultra-peer reachable by all the other ultra-peers in the network. Therefore, in principle, an ultra-peer might have a complete knowledge of the content distribution by contacting all the others ultra-peers and storing information about their content. However, this approach is computationally expensive because no assumption is made about the number of ultra-peers in the network, which might be arbitrarily high. An aspect we are going to investigate is the way to select the neighbours in SPINA, and if a possible criterion based on the content of the neighbours might affect or improve the effectiveness of the query routing strategy dictated by the adopted weighing framework. We are going to investigate experimentally the efficacy of this criterion. Moreover, this experimental evaluation allows the efficacy of the designed architecture and of the adopted ranking scheme to be evaluated in a real setting.

Another issue related to the limited knowledge an ultra-peer has about the others, is the selection of the best neighbour to which the query will be routed. As previously mentioned, the experimental results reported in [11] showed how the first peer or ultra-peer selected and, therefore, visited gives a quite large proportion of recall and that subsequent ultra-peers little contribute. The problem is that routing the query to the nearest neighbour in terms of network load is not necessarily be the best choice to find the best ultra-peer of the network in terms of content. This is clearly due to the degree of decentralization of the considered topology and to the lack of a network structure provided, for example, by a DHT. Indeed another issue is to find a way to enlarge the horizon of the SPINA neighbours, without overloading the network.

3.3 The final ranked result list

An open issue is the strategy to collect the results obtained by the P2P search. Indeed the end-user may be interested in a final list of documents ranked by relevance. The problem is that the score assigned to the documents by the local search engines of the peer might be related to the collections where the document is placed. If the score of the documents stored in different collections will be not comparable, a straightforward comparison of these scores would have no meaning. The same problem would exist also for the resources at the higher levels. For instance, the score assigned by an ultra-peer to the peers of its group might be not comparable with the score assigned to the peers of another group by the ultra-peer they refers. In this case the score would be comparable in the scope of the peer group. A theoretical investigation of this issue might be helpful to understand if this problem exists for the adopted weighing framework.

As regards the results, at the present time the identifier of the document is returned, together with the information needed to contact the peer which stores the document of interest, thus allowing the user to access the entire document. The current implementation does not exploit the logical structure of the documents (which are then an instance of semi-structured data) thus exploiting the information provided by an application of XML. A semi-structured document can be thought as a hierarchy [4]: the whole document, metadata sections and multimedia fragments. The result returned to the user may then consist of several entry points to a same document, corresponding to structural elements, whereby each entry point is weighted according to how it satisfies the query. Then, the estimation of the relevance of a document may be based on the aggregation of the estimated relevance of its related components. Therefore, if we will not require Eq. 2 to be valid for $z > 1$, thus considering documents as structured by sub-resources, the adopted ranking scheme might be used to compute the weight of the entire document by aggregating the weight of its component.

4 Conclusions

In this paper the current status of the implementation of the SPINA software architecture has been described. The way SPINA was designed guarantees independence of the underlying network infrastructure and the media of the document stored in the peer collections. The latter feature helps to extend the presently implemented functionalities in order to search music by content across an hybrid unstructured hierarchical network.

Besides the issues previously mentioned, other aspects will be investigated, for instance the churn of the network, i.e. the dynamics of peer participation, and the policy according to which the groups of peers are formed — i.e. to which ultra-peer a peer is associated. Might this policy be based on the content of the peers or only designed to guarantee a balanced allocation of the peers, in order to avoid an excessive load for the ultra-peers? These issues have to be tackled in order to take into consideration the dynamics of P2P networks.

Acknowledgements

The authors are grateful the reviewers for the comments and suggestions and to Maristella Agosti, Giorgio Maria Di Nunzio, Riccardo Miotto and Nicola Orio for the fruitful discussions on the topic of this paper. The work reported in this paper has been partially supported by the SAPIR project, as a part of the Information Society Technologies (IST) Program of the European Commission (Contract IST-045128).

References

1. JXTA JXSE Project. [Online]. https://jxta-jxse.dev.java.net/ [last visited 2008, May 22], November 2007.
2. Apache Lucene project. [Online]. `http://lucene.apache.org/` [last visited 2008, May 22], May 2008.
3. M. Agosti, E. Di Buccio, G. M. Di Nunzio, N. Ferro, M. Melucci, R. Miotto, and N. Orio. Distributed Information Retrieval and Automatic Identification of Music Works in SAPIR. In *Proceedings of SEBD-2007*, pages 479–482, June 2007.
4. A. Broder, Y.S. Maarek, M. Mandelbrod, and Y. Mass. Using XML to Query XML – From Theory to Practice. In *Proceedings of RIAO*, 2004.
5. E. Di Buccio, N. Ferro, and M. Melucci. Content-based Information Retrieval in SPINA. In *Proceedings of IRCDL 2008*, January 2008.
6. J. Callan, Z. Lu, and W. B. Croft. Searching distributed collections with inference networks. In *Proceedings of SIGIR'95*, pages 21–28, New York, NY, USA, 1995.
7. R. Castiglion and M. Melucci. An evaluation of a recursive weighing scheme for information retrieval in peer-to-peer networks. In *Proceedings of P2PIR'05*, pages 9–16, November 2005.
8. A. Crespo and H. Garcia-Molina. Routing indices for peer-to-peer systems. In *Proceedings of ICDCS'02*. IEEE Computer Society, July 2002.
9. E. Di Buccio and M. Melucci. Utilizing event spaces in distributed Information Retrieval. In *Proceedings of ICTIR'07*, pages 223–232, October 2007.
10. J. Lu and J. Callan. Full-text federated search of text-based digital libraries in peer-to-peer networks. *Information Retrieval*, 9(4):477–498, 2006.
11. M. Melucci and A. Poggiani. A study of a weighting scheme for information retrieval in hierarchical peer-to-peer networks. In *Proceedings of ECIR'07*, pages 136–147, April 2007.