Content-based Information Retrieval in SPINA

Emanuele Di Buccio*, Nicola Ferro* and Massimo Melucci*

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

I. Introduction

The basic rationale of the P2P paradigm is that the processes of a computer system 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 [1].

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 the topology of the network is dynamic, that is, peers can join and leave.

A network can be either structured or unstructured. The former are based on a predefined structure — Distributed Hash Tables 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. This kind of networks allows for 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 some peers, called ultra-peers, with previously established attributes — for example with more CPU, bandwidth or disk than the others — 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 network is called *hierarchical*. The ultra-peers communicate to form the backbone of hybrid decentralized networks. The presence of the hierarchy allows the number of messages to be reduced during query routing [2]. Hierarchical unstructured networks required lower degree of collaboration than structured networks. Indeed, 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" [1].

When P2P networks federates IR systems which provide, for example, search functionalities to the users of a DL, 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.

In order to address the problem of loss of recall, it is our opinion that the design of a P2P-IR system should be done both at a modeling and at an architectural level. Whereas a weighing scheme was proposed in [3] for addressing the design of a P2P-IR system at modeling level, in this paper, the problem will be addressed at an architectural level. A software architecture called SPINA (*Superimposed Peer Infrastructure for iNformation Access*) will be described. While naturally reflecting the proposed weighing framework, SPINA aims at encompassing indexing and retrieval of unstructured documents stored in a P2P network.

The following sections describe the architecture and the weighing model adopted and the current status of the design and the implementation of SPINA.

II. SPINA

The main characteristics of the SPINA software architecture can be summarized as follows [4]:

- it aims at being independent of both the underlying network infrastructure and the media of the documents stored in the network;
- it is focused on exchanging statistics about the features extracted from the full-content of indexed documents and aggregating the resources according to the hierarchy;
- 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.

A. The Architecture

SPINA aims at being independent of the underlying network infrastructure: different network topologies are supported, ranging from unstructured, to hybrid and hierarchical. As depicted in figure 1, SPINA "superimposes" different logical layers over an existing P2P infrastructure. In the example depicted in the figure, three levels are considered: starting from the "lower" one, we can enumerate the (1) document, the (2) peer and the (3) ultra-peer level. The way the SPINA software architecture was designed allows the approach to be

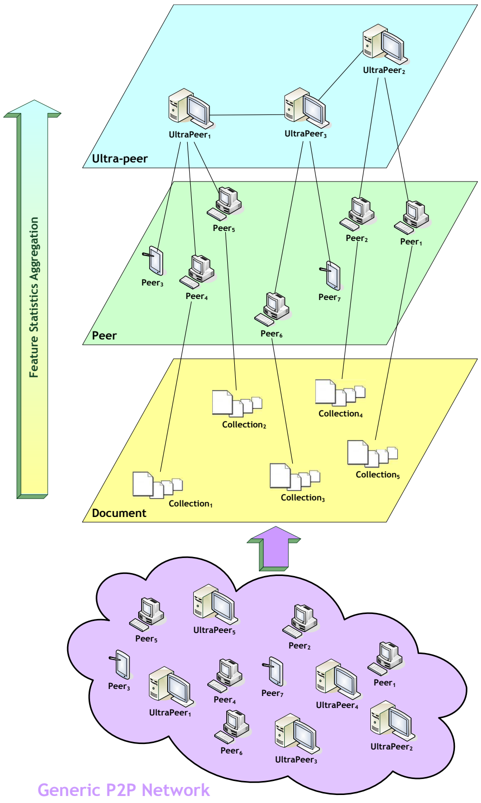

Fig. 1. SPINA Layers.

generalized for an arbitrary number of levels. The current implementation, as described in section III, is based on a three-level hierarchy, as the one depicted in Figure 1. Another feature depicted in the figure is the hierarchical nature of the considered architecture: each group of peers refers to one and only one ultra-peer.

B. 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 fragments 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 figure 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. Indeed, 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 Figure 1, the last level requires an "horizontal" aggregation: indeed 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.

C. The Weighing Framework

An innovative aspect of this architecture is that the algorithms which rank resources at the different levels – documents, peers and ultra-peers – are based on probabilistic models: resources are ranked according to their probability of relevance to the query [5]. Using these algorithms, the peers of a group will 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.

In particular, the adopted weighing framework is the one proposed in [3], that is the TWF (*Term Weighted Frequency*) IRF (*Inverse Resource Frequency*). This ranking scheme looks like a TF·IDF scheme, but its components are in turn TWF·IRF schemes which are recursively defined on top of hierarchy of types of peer. Since the resources including features which occur within few resources are top ranked, the framework supports selecting few resources by thus helping minimize bandwidth.

For each level z, the TWF·IRF is defined as follows:

$$w_{i,j,t}^{(z)} = \mathrm{twf}_{i,j}^{(z)} \cdot \mathrm{irf}_{i,t}^{(z)}, \tag{1}$$

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

$$\mathrm{twf}_{i,j}^{(z)} = \left(\sum_{s=1}^{N_j^{(z-1)}} \mathrm{twf}_{i,s}^{(z-1)} \right) \cdot \mathrm{irf}_{i,j}^{(z-1)}, \tag{2}$$

where $N_j^{(z-1)}$ is the number of resources of level $z-1$ in $r_j^{(z)}$, $\mathrm{twf}_{i,s}^{(z-1)}$ is the TWF of the feature i in $r_s^{(z-1)}$. The IRF of a feature i w.r.t. the resource $r_j^{(z)}$ is a generalization of the IDF for the resources at level z.

Eq. 1 and Eq. 2 allow the weight of the feature in a generic resource $r_j^{(z)}$ to be computed. The score of the resource w.r.t. a formulated query Q, can be computed as

$$w_{j,t}^{(z)} = \sum_{i \in Q} w_{i,j,t}^{(z)} \ .$$

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 architecture, "supports" an arbitrary number of levels.

The efficacy of the ranking scheme was investigated in [6]: the obtained experimental results show that the first peer or ultra-peer visited gives the largest proportion of recall, thus confirming the hypothesis that in the top ranked resources the features of the query have the highest discriminative power. One of the benefits of this weighing scheme is that peers and ultra-peers communicate with each other only some data about local indexes to permit peer and ultra-peer ranking; little information is exchanged thus allowing the reduction of the network load. Since the network is a hierarchical one – every peer connects to one ultra-peer – a peer periodically communicates a summary of its own index to one ultra-peer. The summary is a straightforward list of the feature weights for peers.

III. Implementing SPINA

The approach to tackle the P2P-IR problem introduced in the previous section and the need of considering different types of medium, has been the starting point for designing a software architecture whose main entities are depicted in figure 2.

These entities are:

- `Feature`: represents a feature extracted from a multimedia object. For example, a feature may be a keyword extracted from text, a color histogram extracted from an image, and so on. A `Feature` may contain parameters which provide information about it. For example, a parameter of a keyword can be the frequency of the keyword in the resource.
- `Resource`: represents the basic resource carrying information. A `Resource` is different from a multimedia content object: it is a container for features together with their weights. Besides the `Resource`s which describe multimedia objects, SPINA deals with other kinds of `Resource`s: `Peer`, `UltraPeer` and `Query`. In Figure 2, the arrows with continuous line depicted between the interfaces denote an "is a" relationship. A `Peer` "is a" `Resource`. A `Query` "is a" `Resource`. An `UltraPeer` "is a" `Peer` which "is a" `Resource`.
- `Peer`: represents a peer and may contain `Resources` which represent the multimedia content held by the peer. Furthermore a `Peer`:
    - is capable of ranking its own `Resource`s with respect to a given `Query`;
    - may have neighbouring `Peer`s;
    - is capable of ranking its own neighbours with respect to a given `Query`;

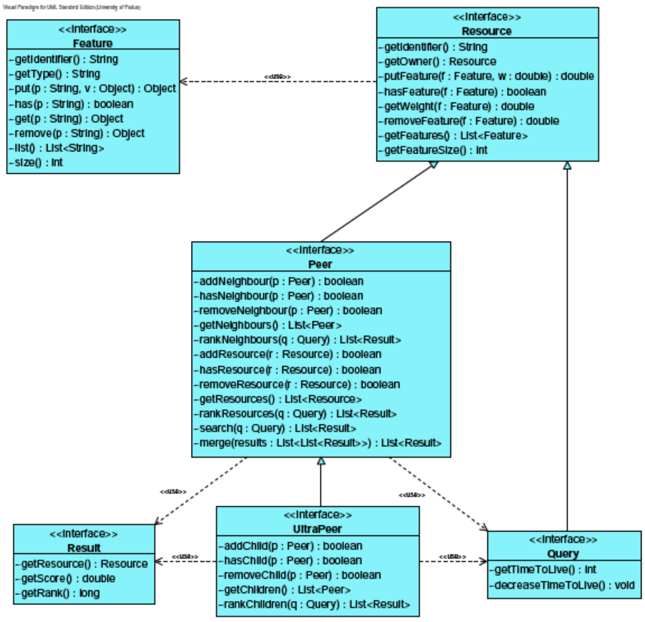
Visual Paradigm for UML Standard Edition(University of Padua)
<<Interface>>
Feature
-getIdentifier() : String
-getType() : String
-put(p : String, v : Object) : Object
-has(p : String) : boolean
-get(p : String) : Object
-remove(p : String) : Object
-list() : List<String>
-size() : int
<<use>>
<<Interface>>
Resource
-getIdentifier() : String
-getOwner() : Resource
-putFeature(f : Feature, w : double) : double
-hasFeature(f : Feature) : boolean
-getWeight(f : Feature) : double
-removeFeature(f : Feature) : double
-getFeatures() : List<Feature>
-getFeatureSize() : int
<<Interface>>
Peer
-addNeighbour(p : Peer) : boolean
-hasNeighbour(p : Peer) : boolean
-removeNeighbour(p : Peer) : boolean
-getNeighbours() : List<Peer>
-rankNeighbours(q : Query) : List<Result>
-addResource(r : Resource) : boolean
-hasResource(r : Resource) : boolean
-removeResource(r : Resource) : boolean
-getResources() : List<Resource>
-rankResources(q : Query) : List<Result>
-search(q : Query) : List<Result>
-merge(results : List<List<Result>>) : List<Result>
<<use>>
<<use>>
<<Interface>>
Result
-getResource() : Resource
-getScore() : double
-getRank() : long
<<use>>
<<Interface>>
UltraPeer
-addChild(p : Peer) : boolean
-hasChild(p : Peer) : boolean
-removeChild(p : Peer) : boolean
-getChildren() : List<Peer>
-rankChildren(q : Query) : List<Result>
<<use>>
<<Interface>>
Query
-getTimeToLive() : int
-decreaseTimeToLive() : void

Fig. 2. SPINA API.

    - being a `Resource`, may be owned by another peer which actually is an `UltraPeer`;
    - being a `Resource`, is capable of providing information about its `Feature`s;
    - is capable of answering a `Query`.
- `UltraPeer`: represents an ultra-peer. An `UltraPeer` manages and organizes a group of `Peer`s and is capable to rank them with respect to a given `Query`. Being a `Peer`, it inherits all the `Peer` properties.
- `Query`: represents a query. It is a `Resource` and is characterized by a Time To Live (TTL), which denotes the maximum number of hops for which the `Query` can be forwarded.
- `Result`: represents a result of a `Query`. A `Peer` — or an `UltraPeer` — replies to the `Query` by returning a list of `Result`s, where each `Result` is a triple: a `Resource`, the weight assigned to the `Resource` according to the adopted weighing scheme, and the consequent rank of the `Resource` in the returned list.

As previously mentioned, the arrows with continuous line indicate an "is a" relationship. In figure 2 also arrows with dashed line are depicted. The dashed arrows indicate a relation of "use" between the different entities. Indeed, a `Resource` is a container of `Features` "used" to represent information about the content of the multimedia object the `Resource` represents. `Peers` and `UltraPeers` "use" a `Query` to do a local or a P2P search and "use" `Results` in order to answer to a formulated `Query`.

At the present time the SPINA software architecture implementation is underway. The functionalities already implemented allows for content-based retrieval of collections of textual documents in a P2P network. The next subsections provide some details about the present implementation.

A. Peer Communication

The JXTA Technology was chosen for implementing the communication among peers, particularly the java implementation JXTA JXSE 2.5 [7]. “JXTA technology is a set of open protocols that enable any connected device on the network, ranging from cell phones and wireless PDAs to PCs and servers, to communicate and collaborate in a P2P manner” [8]. The implemented architecture, as mentioned in the section II-A, is structured on two levels: peers and ultra-peers level. Peers are able to communicate with the ultra-peers they refer to, send queries or answer to it, and send information about their contents so statistics can be aggregated. Ultra-peers are able to communicate with their neighbours in order to send the aggregated statistics about their content and forward a query or answer to it.

B. Retrieval

As regards to the implementation of retrieval in SPINA, we distinguish the document level — local search — from the peer and ultra-peer level — P2P search.

For the “document level”, Lucene [9] was chosen for implementing indexing. An interface between Lucene and SPINA was realised. This approach allows for independence from the specific engine suitable for the resources at the “document level”. Lucene was not directly and “blindly” used for implementing retrieval: the latter was implemented so that different weighing schemes and different ranking algorithms can be utilized. The main reason for this choice is the need of realizing a flexible infrastructure, which takes into consideration the great variety of devices with different functional capabilities which constitute an heterogeneous environment as a P2P network. At the present time, a ranking algorithm based on Document-At-A-Time (DAAT) strategies was implemented; the latter approach guarantees a smaller run-time memory footprint, a suitable characteristic in an heterogeneous environment. The weighing scheme adopted is the TF·IDF as defined in [10].

Resources have to be selected not only at the document-level, but also at the peer-level and ultra-peer-level. For these “higher” levels – peer and ultra-peer level – also the indexing part has to be implemented. The functionalities of Lucene was adopted also to create the high granularity indexes. At these levels a posting list is associated to each feature, where each element of the list is a resource — peers or ultra-peers according to the granularity of the considered index — and the weight associated to the resource. The weight is computed by the TWF·IRF weighing scheme (see Section II-C). In particular, the instance of this weighing framework proposed in [6] was adopted. At the peer level the TWF of a feature is computed by Eq. 2, and then the weight by applying Eq. 1. The weight of a feature in an ultra-peer is computed by considering only the TWF component obtained by applying Eq. 2 at the ultra-peer level, that is for $z = 3$.

IV. Conclusions and Future Works

In this paper the current status of the design and the implementation of the SPINA software architecture is reported. At the present time, the network infrastructure responsible for the communication between peers has been implemented. Also the functionalities required for local and P2P search have been realised: a weighing scheme based on a probabilistic model and the indexes which store information required to compute the weights and consequently to rank resources at different levels. The implemented functionalities allow to retrieve textual documents by content across a P2P network.

Some questions are still open and will be investigated in the near future. The churn[1] of the network is a first issue that will be tackled: in particular how to manage churn and the policy according to which the groups of peers are formed — i.e. to which ultra-peer a peer is associated. Another aspect is the dynamics due to the change of the contents of the collections store in the different peers: statistics update policies will be matter of future research. Another feature that will be implemented in the next future is the integration of Music IR engines, so that not only textual documents, but also multimedia objects can be retrieved — extending search to distributed audio-visual content is the aim of the SAPIR project [11] which supports part of the research activity described in this paper. Lastly, the strategy by which an ultra-peer merges the results returned by the peers of its group will be investigated.

Acknowledgements

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] J. Lu and J. Callan, “Full-text federated search of text-based digital libraries in peer-to-peer networks,” *Information Retrieval*, vol. 9, no. 4, pp. 477–498, 2006.

[2] H. Nottelmann and N. Fuhr, “Comparing different architectures for query routing in peer-to-peer networks,” in *Proceedings of ECIR 2006*. Springer, 2006, pp. 253–264.

[3] R. Castiglion and M. Melucci, “An evaluation of a recursive weighing scheme for information retrieval in peer-to-peer networks,” in *Proceedings of P2PIR 2006*. New York, NY, USA: ACM Press, 2005, pp. 9–16.

[4] 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*, June 2007, pp. 479–482.

[5] E. Di Buccio and M. Melucci, “Utilizing event spaces for Distributed Information Retrieval,” in *Proceedings of ICTIR’07*, Budapest, Hungary, October 2007, pp. 223–232.

[6] M. Melucci and A. Poggiani, “A study of a weighting scheme for information retrieval in hierarchical peer-to-peer networks,” in *Proceedings of ECIR 2007*, April 2007, pp. 136–147.

[7] (2007, November 07) JXTA JXSE Project. [Online]. https://jxta-jxse.dev.java.net/ [last visited 2008, May 27].

[8] JXTA(TM) Community Projects. [Online]. https://jxta.dev.java.net/ [last visited 2008, May 27].

[9] (2008, May 08) Apache Lucene project. [Online]. http://lucene.apache.org/ [last visited 2008, May 27].

[10] A. Singhal, C. Buckley, and M. Mitra, “Pivoted document length normalization,” in *Proceedings of SIGIR’96*. New York, NY, USA: ACM, 1996, pp. 21–29.

[11] Search In Audio Visual Content Using Peer-to-peer IR. [Online]. http://www.sapir.eu/ [last visited 2008, May 27].

[1] Here the term *churn* denotes the dynamics of peer participation