# A Collaborative Approach for Query Propagation in Peer-to-Peer Systems[1]

Anne Doucet, Nicolas Lumineau

LIP6 Laboratory
University of Paris 6
8, Rue du Capitaine Scott, 75015 Paris, France
FirstName.LastName@lip6.fr

**Abstract.** Sharing resources on a world-wide scale is a current research topic. Nowadays, peer-to-peer architecture is considered as a scalable solution to this issue. However, a lot of problems to extend this architecture for data storage remain open. When information is not highly replicated, localization of nodes storing relevant data becomes essential to avoid covering completely the network. To this purpose, we try to propagate a query towards nodes potentially storing relevant data. Information about nodes relevance is obtained by users' experiences. In an applicative context where "communities of interest" exist, we create logical and semantic links not only to specify the nodes relevant for a community, but also to link communities with a related interest. The proposed pattern has been created with respect to Peer-to-Peer philosophy and so as to consider the evolution of communities.

## 1 Introduction

With the development of Internet, the amount of available resources has drastically increased. This leads to revisit the notion of information access in this context. Techniques used in federated or distributed databases, which were developed to manage data distributed on a limited number of servers, have reached their limits in the context of Internet, and do not support scalability. A new class of distributed architecture, peer-to-peer (P2P) systems [1], is designed to support thousands of nodes, providing thus interesting solutions to scalability. In such systems, nodes are indifferently client and server. A query on P2P systems is based on the propagation from node to node until the number of rebounds between nodes is considered sufficient. The node receiving a query solves it locally, and propagates it to a set of neighbors.

Peer-to-peer systems offer many advantages [9], by allowing to access distributed information without specifying the server containing it. However, the principle of propagation has some drawbacks [6]. A first problem is that the definition of a neighbor of a node is done randomly and does not rely on any particular semantics. Another problem is that propagation generates an important number of messages through the network [10].

In this paper, we propose a solution to reduce the number of messages used during the query propagation, as well as the response time [16]. We consider that a user looking for specific data will first query metadata [11][3], which give the user a description of the available resources. Thus, queries we consider here only concern metadata. They are used as a first step to discover sites storing relevant data. The idea used in our approach is based on using some knowledge to adjust the rebounds. Different knowledge levels can be used: user knowledge (profile, experience, history of past queries, belonging to a community of users having the same kinds of interests, etc.) as well as network knowledge (based on the semantics of the node content [4][5]).

We focus in this paper on the user knowledge, and particularly on his/her belonging to a community of users. We claim that a user interested by meteorology will find relevant data on the servers already and successfully queried by users of the same community (for instance meteorologists) or by users of related communities (for instance climatology). Our

[1] This research is done in the context of the PADOUE project ( http://www-poleia.lip6.fr/padoue) financed by ACI GRID : http://www-sop.inria.fr/aci/grid/public

purpose is to introduce this kind of knowledge into P2P systems, in order to direct as better as possible the search, i.e. to propagate the query directly to a relevant node (i.e. containing relevant data). As P2P systems are very evolutive, we insist in this paper on the dynamicity of our approach.

This approach is inspired by works about collaborative filtering [14][2][15]. In our solution, we do not filter on users, as it is generally the case in other related works, but rather on the community, which makes our work original. Our goal is to discover related communities, which will recommend sites containing relevant information. For this purpose, we label the nodes in order to describe the domain of the data they store. This is similar to the notion of semantic labeling used in Semantic Overlay Network [5] where nodes are logically linked according to their contents. But this approach requires a global hierarchy of concepts to classify all items (data, query, nodes, and connections), while our system only uses information on membership of a community and users feedbacks, which can easily be obtained.

The paper is organized as follows. After defining the notion of community in Section 2, we explain in Section 3 how to model it in order to introduce it into P2P systems. We show how to use the notion of links between communities (several communities have similarities, and can share interests) for query propagation in Section 4. We conclude in Section 5, and present some perspectives for this work.

## 2 Community

We suppose that users are clustered around the nodes of the network, according to their physical localization. For example, a node gathers a set of users belonging to the same organism, or the same department, etc. The notion of community we introduce here, allows to cluster again users linked by the same node, according to more logical features. Users belong to the same community if they are bound to the same node (as we previously mentioned) and if they share the same field (or related fields) of interest. To build the links between users, a community is defined by a small set of themes (ie: keywords) which characterize a field of interest. Figure 1 shows examples of such themes concerning specific domains of sciences such as oceanography, oceanology, hydrology, etc which define the fields of interest of the communities.

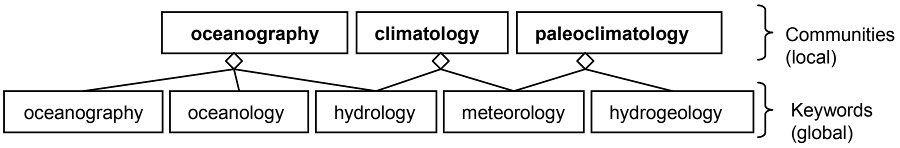

**Fig 1 :** Definition of communities according to themes (keywords). In this example, the community “climatology” is defined for a node by the two keywords “hydrology” and “meteorology”. We note that the set {hydrology, oceanology} may define the community “oceanography” on another node.

We underline the difference between the communities which are local for each node and the use of keywords to define communities. In fact, these themes, established by human experts, are a global resource shared by all nodes and known by all users of the network. This gives the node the autonomy to build its own community and avoids imposing a global definition. We also note that all community definitions are built on a single space of topics, allowing the system to compare them and to create links between them.

# 3 Source of community information

In order to better exploit community experiences, two different kinds of logical links between nodes can be built. On the one hand, we consider *relevance-based links* pointing out potentially interesting nodes for the query execution, and on the other hand, *inter-community links* mapping two similar communities on two different nodes. We present these two kinds of links in the following.

## 3.1 Pattern of relevance-based links

We first consider a set of users which membership of a community of interest is clearly established and easy collectable to avoid discovering it by an elaborate process of users' clustering [13]. For example, a set of researchers in environment may be clustered into several communities: hydrogeologists, climatologists, ecologists, etc. Each user is free to specify his/her community among the list of communities defined on the node (see Figure 2-a) (if there is no relevant community, a user can create a new community).

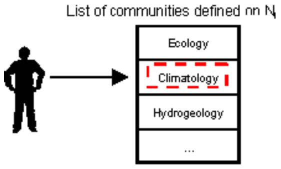

**Fig 2.a**: A user on node $N_i$ specifies his community among the list of available communities

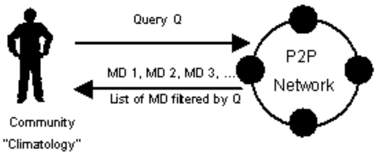

**Fig 2.b**: The user queries the network and the system returns the metadata potentially relevant for the query

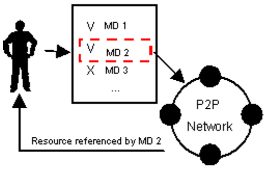

**Fig 2.c**: The user gives his feedback about the returned metadata, and selects the corresponding relevant resources

Moreover, to simplify the process of resource retrieval, we use metadata [7] which describe the resources (data or programs) shared in the network. These metadata allow the user to straightly and quickly build his/her own opinion about the relevance of the resource. Thus, the process of resource retrieval is now treated as a process of metadata localization as shown on Fig 2.b. When resources are discovered, the user gives some feedback (1 if it is relevant, -1 if it is not relevant) on these metadata and selects, among the set of metadata he/she obtained, the relevant resources to load (see Fig 2.c).

For example, a resource as a "pluviometric reading", which contains a set of statement, is described by a metadata defined by a title, a date, an author, a localization, … In this way, users can build structured queries by keywords (for example: "title: pluviometric reading, date:2003, localization: France") to filter relevant metadata.

In the following, we suppose that resources stored on a same node are related to same topics[2]. We thus consider that a node is relevant, if relevant resources (i.e. metadata) have been found on it.

Given $C_{ij}$, the i^th community *defined locally* on the node $N_j$, we can now define the relevance-based links allowing to come out community knowledge. Such semantic link is defined by the following triple (1), stored on node $N_j$ to express that the members of the community $C_{ij}$, consider the node $N_q$ is relevant, with a given aggregated feedback $fa^{q}_{ij}$.

$$\left(N_q,\quad C_{ij},\quad fa^{q}_{ij}\right) \tag{1}$$

[2] This hypothesis is realist in our context where each node is a specific environmental organism which shares its own resources on specific themes.

where in (1), $N_q$ is the IP address of the node $N_q$, $C_{ij}$ is the label of the community $C_{ij}$. We specify that $fa_{ij}^{q} = \text{agg}\left(f_{u_1}^{q},\ldots,f_{u_n}^{q}\right)$ is the feedback achieved by the function agg of aggregation (average) based on individual feedbacks $f_{u_1}^{q},\ldots,f_{u_n}^{q}$ awarded by some users of the community $C_{ij}$ about node $N_q$. (see the right side of figure 3). A community can thus maintain a list of relevant nodes, which are recommended to users belonging to this community. In the same way, this knowledge can be used to avoid querying a node if the community considers it as irrelevant (when the aggregated feedback is negative).

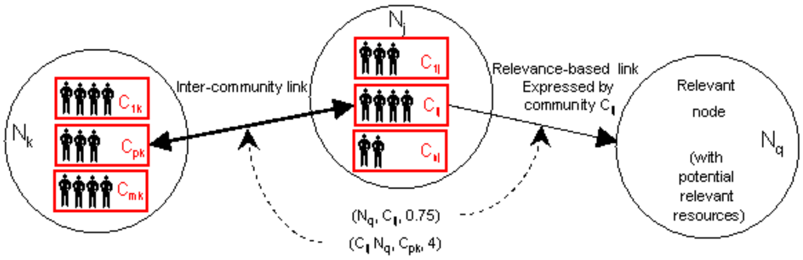

**Fig 3** : Different links between nodes. The relevance-based link between $N_j$ and $N_q$ expresses that $N_q$ is relevant for the community $C_{ij}$ of $N_j$. The inter-community link expresses the similarity between communities $C_{ij}$ and $C_{pk}$ on nodes $N_j$ and respectively $N_k$.

### 3.2 Pattern for inter-community links

Considering only the experiences of a local community is not satisfying. For this reason, we propose to take advantage of experiences of other communities defined on other neighbor nodes. Our idea is to build a link between two nodes on which similar communities exist. Before explaining what similar communities means, we underline the fact that the purpose of these links is not the straight localization of relevant nodes, but only the localization of communities competent to recommend relevant nodes. Our approach of inter-community links gets in two steps: the static creation and the dynamic evolution.

In a first step, we temporarily define the inter-community links between community $C_{ij}$ defined on node $N_j$ and community $C_{pk}$ defined on node $N_k$, by a triple (2) stored on node $N_j$:

$$\left(C_{ij},\quad N_k,\quad C_{pk}\right) \tag{2}$$

where in (2) $C_{ij}$ and $C_{pk}$ are the names of the communities (i.e. themes which specify them) and $N_k$ is the IP address of the node $N_k$. To create these links, we introduce a definition-based similarity which allows to compare the themes defining the communities. In order to define a similar community for the community C, we compare the themes defining C with the sets of themes defining the other communities on a given node. The community having the highest number of common themes with C is considered as similar to C.

Given D(C) which returns the set of themes defining the community C, we say that:

$C_{ij}$ is *def-similar* to $C_{pk}$

if : ₒ $D(C_{ij}) \cap D(C_{pk}) \neq \emptyset$ and, (3)

ₒ $C_{pk} = \underset{c \in \{communiies\ of\ N_k\}}{\arg\max} \left|D\left(C_{ij}\right) \cap D\left(c\right)\right|$ (4)

In other words, we search on $N_k$ a community having at least one common keyword (condition 3) and having the highest number of common themes (condition 4). For example, let us consider the community “climatology” (considered as $C_{ij}$) defined by D(climatology)={hydrology, meteorology} on $N_j$. We consider only communities on $N_k$ having at least the keyword “hydrology” or “meteorology” in their definition. The communities having these two keywords in their definition are considered as *def-similar*. We underline that the *def-similar* definition depends on the original choice of themes. Moreover, these links can be established at the creation of a community or when two nodes become neighbors.

We have shown how to relate communities by two different kinds of links, the relevance-based link and the inter-community link. Figure 3 points out the difference between these two kinds of links. However, this mapping between communities is only static, and does not reflect the evolution of communities. Indeed, we have shown that the knowledge of a community comes out from user feedbacks, which is a dynamic knowledge. We explain in the following section how to take into account the dynamic evolution of communities into the handling of inter-community links.

### 3.3 Handling inter-community links

As experiences of each community evolve, the static inter-community links become obsolete. Their relevance is called into question after being modified many times (consideration of new individual feedbacks). Thus, the use of *def-similar* is not adapted to this evolution. Therefore, we introduce a new definition of the similarity based on the experiences of the community.

As we want to compare dynamically communities, we consider the Pearson correlation coefficient [2]. Indeed, to establish the correlation between $C_{ij}$ and the communities of $N_k$, node $N_j$ sends to node $N_k$ the experiences of the community $C_{ij}$ (i.e. the list of evaluated nodes and their aggregated feedback) to compute all the correlations. The correlation between $C_{ij}$ and a community $C_{xk}$ of $N_k$ is defined by the following formula:

$$\omega(C_{ij}, C_{xk}) = \frac{\sum_{h/N_h \in NC} \left(fa_{ij}^{h} - \overline{fa_{ij}}\right)\left(fa_{xk}^{h} - \overline{fa_{xk}}\right)}{\sqrt{\sum_{h/N_h \in NC} \left(fa_{ij}^{h} - \overline{fa_{ij}}\right)^2 \sum_{h/N_h \in NC} \left(fa_{xk}^{h} - \overline{fa_{xk}}\right)^2}} \qquad (6)$$

where NC is the set of common nodes evaluated by the two communities, and $\overline{fa_{ij}}$ is the average of all aggregated feedbacks provided by the community $C_{ij}$.

Given Ne(C), which returns the set of nodes evaluated by the community C, the use of the correlation between communities allows us to define the experience-based similarity as follows :

$C_{ij}$ is *exp-similar* to $C_{pk}$

if : $o$ $|\,Ne(C_{ij}) \cap Ne(C_{pk})\,| > \delta$ , and (7)

$o$ $C_{pk} = \underset{c \in \{communities\ of\ N_k\}}{\arg\max}\ \omega(C_{ij}, c)$ (8)

Condition (7) allows considering this similarity only if there are enough common evaluated nodes between $C_{ij}$ and $C_{pk}$. If the comparison is well-founded, we update the inter-community link with the community specified by (8). Thus, we keep the links created according to the definition of the community until the set of common evaluated nodes is under a threshold $\delta$ .

However, calculating *exp-similarity* again for each new feedback is not interesting, because the number of computations used for this task is not justified to express the light evolution of the community knowledge. We introduce in our definition of inter-community link a parameter of freshness. An inter-community link between the community $C_{ij}$ on $N_j$ and $C_{pk}$ on $N_k$ with a freshness t, is defined by the quadruplet (9), stored on $N_j$ :

$$\left(C_{ij},\ N_k,\ C_{pk},\ t\right) \qquad (9)$$

where $t \in \{1,\ldots,\theta\}$, with θ being a parameter used as initial value of freshness fixed in such a way that the system does not continuously cast doubt on the relevance of the links.

To sum up, as long as a community C does not have any experiences, the inter-community similarity with C is based on *def-similarity*; otherwise, the system periodically establishes the community the closest of C with the *exp-similarity* according to the evaluated nodes and their aggregated feedback.

## 4 Query propagation and exploitation of community information

Now that the notions of static and dynamic links between communities are defined, we focus on how to exploit all these information stemming from a community. In particular, we will explain to which nodes a query is propagated and why. According to Figure 4 and given the user U belonging to community $C_{ij}$ on the node $N_j$, the query Q is handled by node Nj.

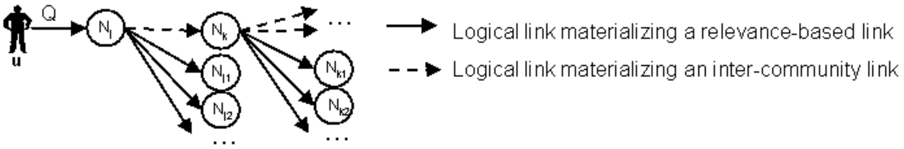
Q
u
$N_j$
$N_k$
$N_{j1}$
$N_{j2}$
$N_{k1}$
$N_{k2}$
...
Logical link materializing a relevance-based link
Logical link materializing an inter-community link

**Fig 4**: Illustration of query propagation

In a first step, the system returns the nodes $N_{j1}$, $N_{j2}$, etc, specified by some relevance-based links of $C_{ij}$ i.e. nodes obtained by filtering the stored triplets (1) on $N_j$ having a positive aggregated feedback. Then, node $N_j$ sends the query Q to these nodes supposed to store relevant resources. Moreover, the inter-community links stored on $N_j$ specify that there are a community $C_{pk}$ on $N_k$ related to $C_{ij}$. The query Q is sent on $N_k$ and will not be executed on it, but on the nodes that the community $C_{pk}$ considers relevant ($N_{k1}$, $N_{k2}$, … obtained by relevance-based link of the community $C_{pk}$). According to the principle of peer-to-peer, the node $N_k$ treats Q as its proper query. Therefore, Q will be straightly propagated towards relevant nodes, in order to find relevant resources.

## 5 Conclusion and future work

Information retrieval in large distributed systems introduces new challenges for database technology. Peer-to-peer architectures offer interesting solutions to support scalability. However, they must be adapted to increase their functionalities and their performance. We propose in this paper to improve the performance of peer-to-peer systems by introducing knowledge in the process of query propagation. This knowledge is used to select the nodes to which the query will be redirected. The kind of knowledge we introduce here is the belonging of a user to a community, and the links which can be established between related communities. This knowledge is both static and dynamic. We have shown how to define and how to use these links to direct query propagation.

This work has been done in the framework of the PADOUE project [12], which aims at building a complete system for sharing heterogeneous and distributed environmental information and in which we exploit the experience of scientific communities. Further work is still to be done. We are currently validating our proposition by integrating this knowledge into the route tables of a peer-to-peer system. An important perspective is to consider other kinds of knowledge. In this work, we focused on community knowledge, but we intend to take into account other knowledge levels, such as user profile, or network knowledge. Our purpose consists of exploiting complementary sources of semantic to smartly extract information from a peer-to-peer system.

## References:

1. Aberer, K., Hauswirth, M., Peer-to-Peer Information Systems: Concepts and Models, state-of-the-art, and Future Systems, Tutorial IEEE ICDE, 2002.
2. Breese, J.S., Heckerman, D., Kadie, C., Empirical analysis of predictive algorithms for collaborative filtering. In Proc. of the 14th conference on uncertainty in artificial intelligence, pp. 43-52, Madison, Wisconsi, July 1998.
3. Coulondre, S., Libourel, T., Spéry, L., Metadata and GIS : a classification of metadata for Gis, Gis Planet'98, International Conference and Exhibition on Geographic Information, Lisbon, Portugal, September 1998
4. Crespo, A., Garcia-Molina, H., Routing Indices for Peer-to-Peer Systems, In Proc. of the 22th International Conference on Distributed Computing Systems (ICDCS) Vienna, Austria, 2002
5. Crespo, A., Garcia-Molina, H., Semantic Overlay Networks for P2P Systems, submitted for publication, (http://www-db.stanford.edu/peers) 2003
6. Daswani, N., Garcia-Molina, B., Yang, B., Open Problems in Data-Sharing Peer-to-Peer Systems, In Proc. of the 9th International Conference on Database Theory (ICDT), Siena, Italy, 2003
7. Galhardas, H., Simon, E., Tomasic, A., A framework for classifying scientific metadata, AAAI Workshop on AI and Information integration, Madison, Wisconsin, August 1998.
8. Golberg, K., Roeder, T., Gupta, D., Perkins, C., Eigentaste: A Constant Time Collaborative Filtering Algorithm, Technical report IOER and EECS Departments, University of California, Berkley, August 2000.
9. Gribble, S., Halevy, A., Ives, Z., Rodrig, M., Suciu, D., What Can Peer-to-Peer Do for Databases, and Vice Versa?,. In Proc. of the 4th International Workshop on the Web and Databases (WebDB '2001), Santa Barbara, California, May 2001.
10. Jovanovic, M.A., and al, Scalability Issues in Large Peer-to-Peer Networks – A Case Study of Gnutella, Research report, Univ. Cincinnati, 2001
11. Moura, A., Perez, H., Tanaka, A., Metadata model for supporting data extraction from environmental information systems, In Proc of the International Conference On Geographic Information Science, Savannah, Georgia, October 2000
12. PADOUE Project (http://www-poleia.lip6.fr/padoue) multifield project of ACI-GRID: http://www-sop.inria.fe/aci/grid/public
13. Seng, S.H., Han, J., Wang, K., RecTree: An efficient collaborative filtering method, in Proc. The conference on data Warehouse and Knowledge Discovery, (DaWaK), Munich, Germany, 2001
14. Resnick, P, Varian, H.R., Recommender Systems. In Communications of the ACM , Vol 40. N°3, March 1997
15. Ungar, L., Foster, D., Clustering methods for collaborative filtering, In Workshop on Recommender systems at the 15th National Conference on Artificial Intelligence, Madison, Wisconsin, July 1998.
16. Yang, B., Garcia-Molina, H., Improving Search in Peer-to-Peer Systems, In Proc. of the 22th International Conference on Distributed Computing Systems (ICDCS), Vienna, Austria, July 2002