P2PShare: a Social-based P2P Data Sharing System

Fady Draidi, Esther Pacitti, Didier Parigot, Guillaume Verger, Patrick Valduriez, Remi Coletta, Reza Akbarinia, Emmanuel Castanier

P2PShare: a Social-based P2P Data Sharing System[1]

Reza Akbarinia, Emmanuel Castanier, Rémi Coletta, Fady Draidi, Esther Pacitti, Didier Parigot,

Guillaume Verger, Patrick Valduriez

Inria & LIRMM, Montpellier, France

{*Firstname.Lastname@inria.fr or Firstname.Lastname@lirmm.fr* }

Résumé

P2PShare est un système Pair à Pair (P2P) pour le partage à grande échelle de données probabilistes qui s'appuie sur un système de recommandation basée sur le contenu et l'expertise de chaque utilisateur. Il est conçu pour gérer des données probabilistes et déterministes dans un environnement P2P. Il fournit un environnement flexible pour l'intégration de sources hétérogènes, et prend en compte les aspects sociaux pour découvrir des résultats de haute qualité pour les requêtes en privilégiant les données des amis (ou des amis d'amis), qui sont les experts sur les sujets liés à la requête. Nous avons mis en place un prototype de P2PShare à l'aide du logiciel Shared-Data Overlay Network (SON), une plate-forme de développement open source pour des réseaux P2P utilisant les services web, JXTA et OSGi. Dans cet article, nous décrivons les principaux services de P2PShare, par exemple, la diffusion des sujets d'intérêt entre les amis, les requêtes sur les contenus et les requêtes probabilistes sur les jeux de données.

Abstract

P2PShare is a P2P system for large-scale probabilistic data sharing that leverages content-based and expert-based recommendation. It is designed to manage probabilistic and deterministic data in P2P environments. It provides a flexible environment for integration of heterogeneous sources, and takes into account the social based aspects to discover high quality results for queries by privileging the data of friends (or friends of friends), who are expert on the topics related to the query. We have implemented a prototype of P2PShare using the Shared-Data Overlay Network (SON), an open source development platform for P2P networks using web services, JXTA and OSGi. In this paper, we describe the demo of P2PShare's main services, e.g., gossiping topics of interest among friends, key-word querying for contents, and probabilistic queries over datasets.

Mot-clés: Pair à Pair, Recommendation, Base de Données Probabiliste

1. Introduction

P2PShare is a P2P system for large-scale probabilistic data sharing, particularly in scientific communities. It takes into account heterogeneous data, and leverages content-based and expert-based recommendation for discovering the data relevant to queries. It is composed of three main components:

- **ProbDB** (http://probdb.gforge.inria.fr) [1] is a probabilistic database system built on top of a classical DBMS. Instead of directly modifying the DBMS and adding "native" primitives to it, we have chosen to implement ProbDB on top of the DBMS, and thus to be able to change the underlying DBMS with a slight programming effort.
- **WebSmatch** (http://websmatch.gforge.inria.fr) [2] is a flexible environment for Web data integration, based on real, end-to-end data integration scenarios (e.g. over public data or scientific data). WebSmatch supports the full process of importing, refining and integrating data sources and uses third party tools for high quality visualization.
- **P2Prec** (http://p2prec.gforge.inria.fr) [3][4] is a social-based P2P recommendation system for large-scale content sharing that leverages content-based and social-based recommendation. The main idea is to recommend high quality documents related to query topics, by exploiting friendship networks.

[1] Work partially funded by the DataRing project of the French ANR.

P2PShare is organized as a P2P social network and combines the services of the three above components. The scientific data are stored at peers, without any central administration. Using the recommendation service of P2Prec, a user (a peer) is able to build its list of expert friends, and discover the relevant data provided by its friends. WebSmatch is used for integrating heterogeneous datasets, and ProbDB is utilized for storing and querying the probabilistic or deterministic datasets shared in the network.

We have implemented a prototype of P2PShare using SON[2], an open source development platform for P2P networks. With SON, we can easily provide a highly modular architecture for integration of various data types. As P2Prec was already built on top of SON [4], we integrated WebSmatch and ProbDB to the architecture.

In the rest of this paper, we first describe an application scenario, and then present P2PShare's architecture. Finally, we introduce the demonstration.

2. Application Scenario

As a motivating application for P2PShare, we consider a social network of scientists interested in sharing the scientific data (documents, annotations, datasets, etc.) they store locally in their databases. The scientists typically store different kinds of data including: papers and reports they have written, articles they have downloaded, datasets they have used or produced for scientific experiments, etc. In addition, they annotate the documents they have with tags, ratings, etc. The goal of this social network is that each user (peer) can discover data provided by others and send queries to them. For instance, one may want to know which peers are expert in a topic and get all the scientific data highly rated by those experts on that topic. Or another peer may look for the best datasets used by others for some kinds of experiments.

Figure 1 shows an example of the different relations needed to store the scientific data for this social network. The relation *Documents* (doc_id, file_name) stores the id and address of documents (e.g. papers, reports, and articles) which are stored as files. The relation *Datasets* (dataset_id, location) stores the id and location of the datasets. The datasets may contain probabilistic or deterministic (ordinary) data. In the *Datasets* table, the *location* attribute will be the name of the database that keeps the data. The relationship between the documents and datasets can be specified explicitly by the researchers or be inferred by the system (from the existing relationships). This relationship is shown in the relation *Related_Datasets* (doc_id, dataset_id, probability).

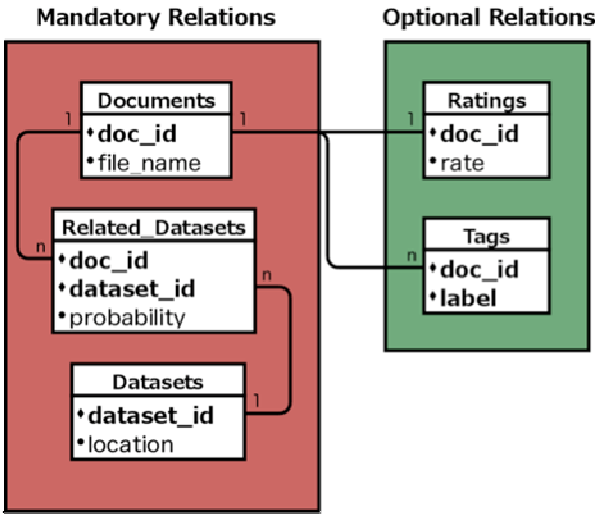

Figure 1. Relations between documents and datasets

[2] http://www-sop.inria.fr/teams/zenith/SON

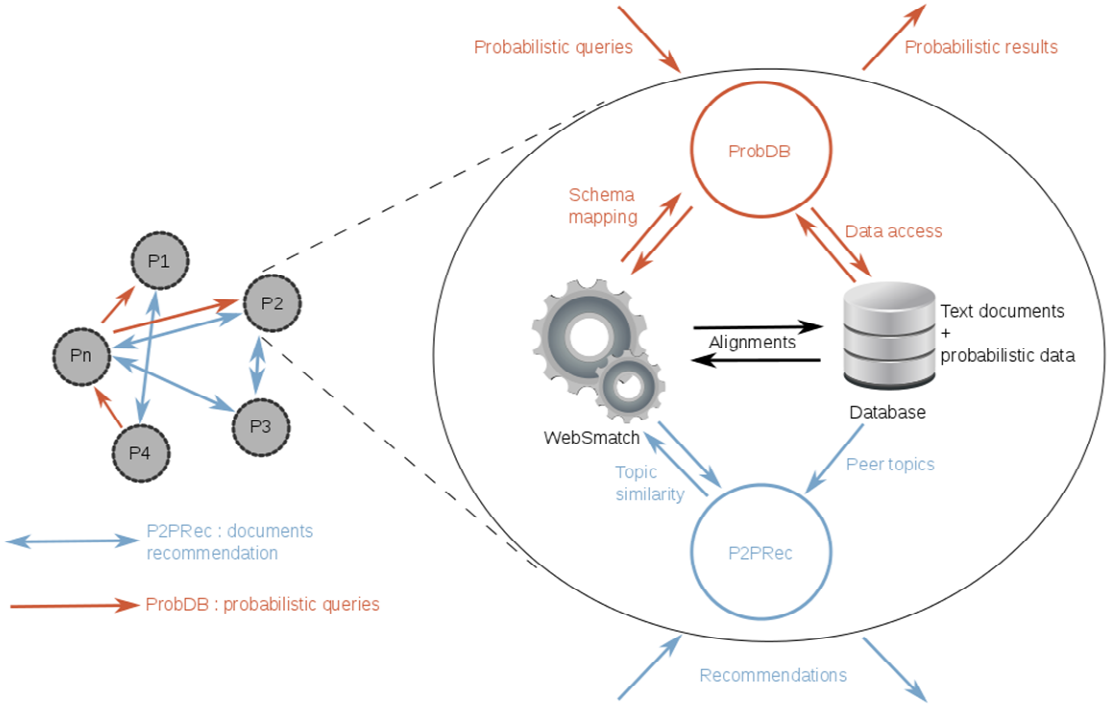

Figure 2. P2PShare Architecture

The confidence on the relationship of documents and datasets is stated in the form of a probability, by using the *probability* attribute. If the probability is set to 1, it means that a researcher has explicitly specified the relationship between the document and the dataset. If the system infers the relationship, then the probability is set according to the semantic distance between the document and the dataset. The semantic distance is computed by WebSmatch by using the combination of several techniques, and normalizing their outputs. The closer the document and the dataset, the lower is the distance value. By using the probability attribute, P2PShare can answer important probabilistic queries issued by researchers, for example: given a document *d* and a probability *p*, return all datasets that are related to *d* with a probability higher than *p*. To answer such queries, P2PShare takes advantage of probabilistic query processing in ProbDB. In order to locate the data that are related to a query in the P2P network, P2PShare uses the recommendations given by P2PRec.

3. Prototype Architecture

P2PShare has been developed on top of the SON platform. SON is based on a set of basic concepts for developing and deploying multiple services (e.g. directory, query, summary or recommendation) in a simple and effective way. It is an open source development platform for P2P networks using web services, JXTA and OSGi. SON combines three powerful paradigms: components, SOA and P2P. To provide weak coupling between system entities, components communicate by asynchronous message passing. To scale up, we rely on a decentralized organization based on a DHT for publishing and discovering services or data. In terms of communication, the infrastructure is based on JXTA virtual communication pipes, a technology that has been extensively used within the Grid community.

Using SON, the development of a P2P application is done through the design and implementation of a set of components. Each component includes a non-functional code that provides the component services and a code component that provides the component logic (business code). The complex aspects of asynchronous distributed programming (non-functional code) are separated from code components. The Component Generator (CG) automatically generates this non-functional code from a description of services (provided or required services) for each component. This CG component is not present at the execution of the SON infrastructure. SON is implemented in Java on top of OSGi components that provide all basic services for the lifecycle of our components, in particular,

the deployment services. The launching of a SON application is defined through an OSGi configuration, which describes the application components.

We developed P2PShare as a SON application with three components (see Figure 2): the WebSmatch component to extract the keywords of a document, the ProbDB component for the management of probabilistic databases and the P2Prec component for the recommendation process.

With P2Prec we exploit the social (friendship) connections between participants to support data location in dynamic groups of participants. The main idea is to recommend high quality documents related to query topics hold by useful friends (of friends) of the users, by exploiting the friendship network. To exploit friendship links, we rely on Friend-Of-A-Friend (FOAF)[3] descriptions. Our recommendation model relies on a distributed graph, where each node represents a user (peer) labeled with the contents it shares and its topics of interests. The topics each peer is interested in are automatically calculated by analyzing the documents the peer holds. Peers become relevant for a topic if they hold a minimum number of highly rated documents on this topic. A peer *p* becomes useful to a peer *q*, if the number of their common topics of interest is higher than a threshold. To disseminate information about relevant peers, P2Prec relies on gossip algorithms that provide scalability, robustness, simplicity and load balancing.

ProbDB is responsible for processing queries over probabilistic data. It is built on top of a classical Database management system. When a query is received by ProbDB, it is analyzed and probabilistic keywords are extracted. Then classical (non probabilistic) sub-queries are sent to the DBMS that process them and returns intermediate results. Afterwards, probabilistic functions are applied to the intermediate results, and the final results are returned to the user. In its current version, ProbDB is built on PosgreSQL, but can easily be adapted to work on other DBMSs, for instance MySQL.

Each peer can contain different schemas for describing its database. To be able to reformulate queries, P2PShare uses the schema matching done by WebSmatch. Schema matching is hard as we must deal with both structural heterogeneity (differences in formats, keys, etc.) and semantic heterogeneity (synonyms, homonyms, hypernyms, ambiguous names, etc.) among the metadata. The approach of WebSmatch is not to develop new matching techniques, but rather to combine the state of the art techniques. In its current version, WebSmatch supports dozens of string metrics (from the Second Strings project[4]), dictionary measures from Wordnet[5], Information Retrieval techniques (using the Lucent library[6]) but also some structure and instance-based measures. WebSmatch has two phases: a learning phase that produces a dedicated schema matcher and a matching phase in which the dedicated matcher is used over new input schemas. The learning process can use preferences in terms of precision and recall and some expert-provided correspondences.

4. Demonstration

To demonstrate the P2PShare systems, we use it for implementing the social network of scientists that we described in Section 2. We install P2PShare on each peer of the social network. Each peer will be started with an initial state containing a list of documents, list of datasets, and a list of friends. For the list of documents, we use the Ohsumed document corpus. It is a set of 348566 references from MEDLINE, the on-line medical information database, consisting of a large set of medical publications over a five-year period (1987-1991). We show how the application works, from the global initialization to the utilization by an end-user.

Initialization. As the demonstration starts, an entry point, and several virtual peers (e.g. 30 peers) are created. Each peer is given an initial friend-of-a-friend (FOAF) file, which determines its friends and some information about them. When connecting a new peer to the network, we show how it gets initial information in its FOAF file in two cases: (1) it has already joined the network in the past (i.e. it knows other peers); (2) it connects to the network for the first time.

[3] http://www.foaf-project.org
[4] http://secondstring.sourceforge.net/
[5] http://wordnet.princeton.edu/
[6] http://lucene.apache.org/

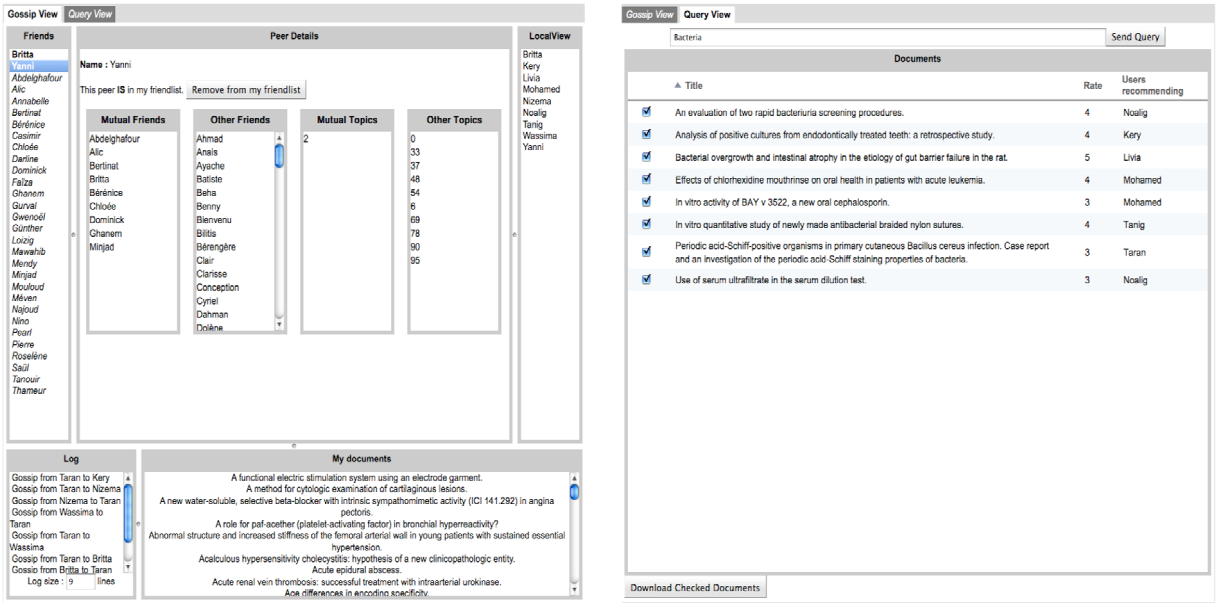

Figure 3. P2PShare GUI

Gossiping. The gossip service is at the heart of P2PShare, and is transparent to the end-user. While peers exchange gossiping messages, the system recommends new friendships to users. For the sake of the demonstration, we developed an interface showing what is internally happening during gossiping. The interface (see Figure 3) shows the current friends of the user, the gossip messages sent and received by the peer, the gossip local-view that permits to find friends, etc. We show how the gossip mechanism notifies the user that other users share the same interests, and asks to add them to her list of friends.

Recommendation-based querying processing. One of the main objectives of P2PShare is to answer queries issued by users looking for documents and datasets. The user is able to send a query for getting documents recommendations from her friends. Each friend may recommend documents depending on the similarity in terms of keywords and the document rates. Figure 3 shows the result returned to the user for an issued query. We show the results of the query for a user who has been in the network for a long time compared to a new user, and compare the accuracy and the number of answers she gets.

Probabilistic querying. Users can also issue queries over the probabilistic data of the systems by using the ProbDB query service. The issued queries are sent to the peers who may be able to partially answer it. When a peer receives a probabilistic query from the ProbDB query service, it processes the whole query and answers directly to the initiator of the query. Depending on the nature of the query, it may need to transfer the query to other pertinent peers and wait for them to answer. Each peer needs to reformulate a global query to its local schema. In order to do so, the WebSmatch component matches the schemas (the tables) of the global query with the local schemas of the user.

References

1 R. Akbarinia, P. Valduriez, G. Verger. Efficient Evaluation of SUM Queries over Probabilistic Data. *Journal of IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 2012 (to appear).

2 R. Coletta, E. Castanier, P. Valduriez, C. Frisch, D. Ngo and Z. Bellahsene. Public Data Integration with WebSmatch. *Workshop on Open Data (WOD)*, 2012.

3 F. Draidi, E. Pacitti, B. Kemme. P2Prec: a P2P Recommendation System for Large-scale Data Sharing. *Transaction on Large-Scale Data- and Knowledge- Centered Systems*, LNCS, 6790(3), 87-116, 2011.

4 F. Draidi, E. Pacitti, D. Parigot, G. Verger. P2Prec: A Social-Based P2P Recommendation System. *20th ACM Conference on Information and Knowledge Management (CIKM)*, p. 2593-2596, 2011.

5 M. El Dick, E. Pacitti, R. Akbarinia, B. Kemme. Building a peer-to-peer content distribution network with high performance, scalability and robustness. *Information Systems*, 36(2): 222-247 (2011).