P2Prec: A Social-Based P2P Recommendation System
Fady Draidi, Esther Pacitti, Didier Parigot, Guillaume Verger

To cite this version:

HAL Id: lirmm-00640886
https://hal-lirmm.ccsd.cnrs.fr/lirmm-00640886v2
Submitted on 27 Nov 2011

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
P2Prec: A Social-Based P2P Recommendation System

Fady Draidi
LIRMM
Montpellier, France
Fady.Draidi@lirmm.fr

Esther Pacitti
LIRMM
Montpellier, France
Esther.Pacitti@lirmm.fr

Didier Parigot
INRIA
Sophia Antipolis, France
Didier.Parigot@inria.fr

Guillaume Verger
INRIA
Montpellier, France
Guillaume.Verger@inria.fr

ABSTRACT

P2Prec 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 and contents held by useful friends (of friends) of the users, by exploiting friendship networks. We have implemented a prototype of P2Prec 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 P2Prec’s main services (installing P2Prec peers, initializing peers, gossiping topics of interest among friends, key-word querying for contents) using our prototype implemented as an application of SON.

Categories and Subject Descriptors
H.3.4 [Systems and Software]: Distributed systems, H.3.3 [Information Search and Retrieval]: Search process, H.3.1 [Content Analysis and Indexing]: Indexing methods.

General Terms
Algorithms, Experimentation

1. INTRODUCTION

The general problem we address is large-scale content sharing for on-line communities. Consider, for instance, a scientific community (e.g., in bio-informatics, physics or environmental science) where community members are willing to share large amounts of documents (including images, experimental data, etc) stored in their local servers. Assume also that they don’t want to lose control over their data at a central site. A promising solution is to organize community members in a peer-to-peer (P2P) overlay network, with the advantages of decentralized control, peer autonomy and scalability.

Locating contents based on content ids in a P2P overlay network is now well solved (see e.g. [5]). However, the problem with current P2P content-sharing systems is that the users themselves, i.e., their interest or expertise in specific topics, or their rankings of documents they have read, are simply ignored. In other words, what is missing is a recommendation service that, given a query, can recommend relevant documents by exploiting user information.

Sinha et al. [10] have shown that in general users prefer the advice coming from known friends in terms of quality and trust because usually users trust their friends’ advice. In most of existing P2P solutions, friendship links are extracted from user’s behaviors [2], or are established based on explicit trust declaration [9]. To enrich these solutions, we exploit the fact that users who store similar contents are potential friends. Therefore, our solution leverages between content-based and social-based recommendations over a P2P overlay.

P2Prec is a social-based P2P recommendation system for large-scale content sharing [4]. The main idea is to recommend high quality documents related to query topics and contents held by useful friends (of friends) of the users, by exploiting friendship networks. Our recommendation model relies on a distributed graph, where each node represents a user (peer) labelled with the contents it stores 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 certain number of highly rated documents on this topic. A peer becomes useful to a peer if its relevant topics are overlapped. To exploit friendship links, we rely on Friend-Of-A-Friend (FOAF) descriptions (http://www.foaf-project.org). To disseminate information about relevant peers, we rely on gossip algorithms that provide scalability, robustness, simplicity and load balancing. In addition, we propose an efficient query routing algorithm that selects the best peers to recommend documents based on users’ useful friends and query topics. At the query’s initiator, recommendations are selectively chosen based on similarity, rates and popularity or other recommendation criteria.

We have implemented a prototype of P2Prec by using the Shared-Data Overlay Network (SON) (http://www-sop.inria.fr/teams/zenith/SON), an open source development platform for P2P networks using web services, JXTA and OSGi (http://www.osgi.org). SON components communicate by asynchronous message passing to provide weak coupling between system entities. To scale up and ease deployment, we rely on a Distributed Hash Table (DHT) for publishing and discovering services or data.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

CIKM ’11, October 24–28, 2011, Glasgow, Scotland, UK,
Copyright 2011 978-1-4503-0717-8/11/10...$10.00

1 Work partially funded by the Dataring project of the Agence Nationale de la Recherche.
In this paper, we describe the demo of P2Prec’s main services (installing and initializing P2Prec peers, gossiping topics of interest among friends, key-word querying for contents) using our prototype implemented as an application of SON.

2. OVERVIEW OF P2Prec
Centralized systems recommender systems (RS) rely on the ratings that users provide [1]. The advents of Web2.0 tools and the growing popularity of online social networks have led to the development of social-based RS that use users’ social data such as friends, trust, etc. to provide recommendations [9]. These systems exploit the preferences and relations of users’ friends (of friends) [2] or the trust relations [8] between users to aggregate the neighbors of each user. Then the recommendations are computed based on the ratings that have been given by those neighbors.

P2Prec’s recommendation model is expressed based on a graph $G = (D, U, E, T)$, where $D$ is the set of shared documents, $U$ is the set of users $u_1,...,u_n$ corresponding to autonomous peers $p_1,...,p_n$, $E$ is the set of edges between the users such that there is an edge $e(u,v)$ if users $u$ and $v$ are friends, and $T$ is the domain of topics. Each user $u \in U$ is associated with a set of interests $T_u \subset T$, and each of these interests are of interest for $u$’s friendship network. A user $v$’s friendship network is $\Pi$.

Once a query is submitted by $u$, it is forwarded to $u$’s top-k useful and trustful friends. When a query is received at any peer, it is again redirected to its top-k useful and trustful friends, until TTL is reached. Each user $v$ that received the query provides recommendations to $u$. The response to a query $q$ is a recommendation that has been provided in a ranked list, based on a function that ranks each document according to its relevance with $q$, its popularity, the similarity and trust between $q$’s initiator and responder $v$.

3. P2Prec IMPLEMENTATION
With 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 and automatically generated. From a description of a component’s services, the Component Generator (CG) automatically generates the non-functional code. Thus, the programmer does not deal with complex distributed programming aspects.

The basic infrastructure of SON is composed of a Component Manager (CM), a Publishing and Discovery Component (PDC), and a Connection Component (CC). The PDC allows publishing or discovering components on different peers using a DHT. The CC provides connection between remote components on peers. The CM performs the creation of new component instances and the connections between them.

To establish a connection between two components, the CM uses the services description of each component. At run-time, when a component A wants to connect with another component B, it must use the service $\text{ConnectTo}(A, B)$ provided by the CM. As in the CM that created the component, the components are by default connected to the CM. To establish a connection between two components, the CM uses the services description to associate the
To run the P2Prec application, the BS must be started on a given machine (with a given IP address). This IP address will be used as the entry point into the P2Prec network for new peers. At the startup time, a new peer must first identify itself with the BS (connect service) and the BS is going to return the current set of all topics (allTopics service). Then within the local peer's LDA component and the current topics, the topics of each document is computed locally.

After these steps, the peer can start the recommendation steps and documents discovery without any connection with the BS. Indeed, the research of topics of a new document (computeTopic(doc) service) and the computing of topics of a query (computeTopic(query) service) can be made locally with the local peer's LDA component. Depending on the evolution of documents on the P2Prec network, the BS may update the set of topics of documents, and inform the peers by broadcasting this new topic set (using the allTopics service).

4. P2Prec DEMONSTRATION

In this section we describe how the P2Prec services cooperate using scenarios based on the Ohsumed documents corpus [6] that has been widely used in IR. It is a set of 348566 references from MEDLINE, the on-line medical information database, consisting of titles or abstracts from 270 medical journals over a five year period (1987-1991). Our application is lightweight, meaning that no client needs to be downloaded to use it.

In order to couple the P2Prec core, made of OSGi configurations, with the chosen scenarios, we used the Google Web Toolkit (GWT: http://code.google.com/webtoolkit/) to build the user interface. This toolkit allows defining a client/server application written completely in Java that runs in a web browser. It automatically compiles the Java client code into HTML and JavaScript, and easily permits to use Java libraries. Therefore, all graphical interfaces of this demo are made of web pages and run in a classical browser. We show how the application works, from the global installation to the utilization by an end-user.

Installation In order to run a P2Prec peer properly, any user (at a peer) has to connect first to the Bootstrap Server (BS). Therefore we define a place the BS will run on. Every peer in the system will know its IP address. As the BS and any peer offer the same kind of services, we have defined two OSGi configurations for running P2Prec components: one as a BS, and one as a standard peer that will connect to the BS.

Initialization Each peer consists of a LDA part coupled with a Communication part (called P2Prec). As the demonstration starts, the BS is created, and so are several peers (30 of them). Each peer sends some of its documents, which are arbitrarily distributed
among all peers, to the BS to perform LDA on a sample of all documents and to define the set of topics used in the network. Next, the BS informs all connected peers about the topics that are present in the network, and each peer indexes its own documents with the set of topics. Each peer is given an initial FOAF, which determines its friends in the network, and provides it information about them. It can now start gossiping with other peers, and the user belonging to the peer can send queries to discover documents. 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.

The user is able to send a query for getting documents. This is where the query service is needed. The BS informs all connected peers about the topics that are present in the network, and each peer indexes its own documents with the set of topics. Each peer is given an initial FOAF, which determines its friends in the network, and provides it information about them. It can now start gossiping with other peers, and the user belonging to the peer can send queries to discover documents. 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.

The demo with the MEDLINE information database (with a fixed data set) can be downloaded from the P2Prec website (http://www-sop.inria.fr/teams/zenith/p2prec/) with the complete procedure for installing, deploying and running, using OSGi configurations. We also provide a second procedure for building P2Prec applications with any data set, from scratch. But before to be used by end-users, it requires setting up a Bootstrap Server and compute (initialize) the set of topics of the given data set.

5. REFERENCES