Experiences with complex user profiles for approximate P2P community matching

Patrizio Dazzi  
ISTI-CNR  
Pisa, Italy  
p.dazzi@isti.cnr.it

Matteo Mordacchini  
IIT-CNR  
Pisa, Italy  
m.mordacchini@iit.cnr.it

Fabio Baglini  
University of Pisa  
Pisa, Italy  
fabio.baglini@hotmail.it

Abstract—The problem of defining P2P overlays where peers characterized by similar interests are directly connected is currently an important research issue. We have recently proposed a two layer P2P architecture where the first layer exploits a gossip algorithm for the detection of communities of peers characterized by similar interests, while the second layer defines a DHT storing the profiles of the communities detected in the first layer. The DHT is exploited by peers joining the system to find out a community matching its interests. This paper investigates a DHT based approach supporting a similarity based search of user profiles. Our approach exploits Locality Sensitive Hashing to support the similarity indexing. The paper investigates several types of profiles to model user interests and evaluates the indexing mechanisms of the DHT with respect to the different types. Experimental evaluation has been conducted by considering a real data set.

Keywords-distributed clustering; peer-to-peer; user profiles;

I. INTRODUCTION

The first recommender system, dealing with artists and albums, based on the collaborative filtering technique has been proposed in [11]. After this contribution several tools have been proposed and realized, both from academic communities and companies. These solutions achieve a remarkable success for their ability to model the behavior of human exchange information by word-of-mouth, which is a typical behavior of people sharing one or more interests.

Most of these systems are based on centralized architectures, an approach that on one hand simplifies the overall design, but on the other one presents a set of limitations. These limitations include the reduced scalability, the existence of single-point-of-failure and the concentration of data published by the users at a single authority which is potentially able to access to the whole set of data.

The most promising alternatives to current centralized systems are based on peer-to-peer (P2P) architectures. P2P architectures allow the implementation of systems able to dynamically scale and to be resilient to failures of groups of nodes, even significant ones. Moreover, their exploitation avoid the concentration of information in a single entity.

Most of recommender systems, both centralized and P2P, group homogeneous users in communities. This simplify the exchange of pertinent recommendations among users sharing common interests.

The creation of a P2P system able to support the aggregation of users in communities, identified on the basis of common interests, has been the subject of a number of recent research works carried out by different research groups. These groups belong to different scientific communities and the proposed works start from different points of view. However, the common idea of all the proposed works is to associate with each user a profile, modeling user’s interests, to exploit in order to aggregate them in communities.

In a precedent position paper of ours [6], we proposed a system structured according to two layers. The first one supports the creation and the maintenance of self-emerging user communities, by performing a continuous exchange of information among users. The other layer is devoted to the maintenance of a distributed index of existing communities. It should allow users to cheaply (in terms of transferred data) discover communities pertinent to their interests, even when does not exist any community that exactly matches its profile.

In this paper we aim at realizing such a distributed index. To allow it to support the mentioned features, the index we propose is P2P, supports similarity searches and supports different ways to represent the user profile.

We started from a Distributed Hash Table (DHT). This kind of network provide the functionalities of a distributed data index, with an effective load balancing strategy, a careful management of the disconnection of the nodes and a logarithmic complexity routing both in the number of hops required to resolve a query and in the size of the routing tables.

However, DHTs are not designed to support similarity searches. Indeed, as with non-distributed hash tables, each value is calculated using an identifier associated with the application of a function designed to distribute the results, avoiding to preserve the locality of the original data. Therefore, in order to achieve the functionalities required by our envisioned distributed index, DHTs have to be extended with mechanisms that allow to perform not-exact searches. To do this, we exploit a probabilistic approach based on the use of Locality Sensitive Hash Functions (LSH), proposed in [13] to index the user profiles on a DHT. This leads to a drastically reduced resources usage. LSH approximates the exact solution of the problem, issuing a very reduced amount
of queries while maintaining very high quality results.

Besides supporting similarity searches on DHTs, our distributed index is able to represent the interests of users in different ways. In particular, in this paper we present a study on three different characterizations of the user interests. The study is aimed at finding a good trade-off between the accuracy and the network overhead.

The three different types of profiles are based on sets of attributes defined in spaces made of an arbitrary number of dimensions and comparable by using simple similarity metrics. This allows to build profiles from objects of any kind (e.g. text stored by users, log browsing web, metadata about the media, sets of coordinates about places visited, etc.). The three types of profiles are, respectively, a vector of items, an adjacency matrix of items occurrences and a set of connected components extracted profile by profile from the adjacency matrixes.

The index we propose has been studied from the point of view of complexity and evaluated experimentally using a real dataset, consisting of a list of articles of medical literature consulted by users of PubMed [1], where each publication is identified by title, keywords and abstracts. From this dataset were extracted profiles from over 2700 users.

The results obtained by searching for similarities on our proposed DHT index has been compared against the ones obtained for the same queries, indexing the profiles of individual attributes. This evaluation was carried out using various configurations of LSH indexing functions, for all types of profiles.

The remaining of this paper is organized as follows: Section II reviews the state of the art in the field, Section III present the overall architecture of the system describing the two layers it is made of, with a particular focus on the indexing layer. Section IV describes both the three kind of profiles we propose in this paper. Section V present the complexity, both in terms of network usage and storage space required. Section VI shows the experimental results we obtained. Finally, in Section VII we derive our conclusions.

II. RELATED WORK

A. P-RING

P-Ring [5] is a P2P index structure that supports range queries. P-Ring is designed to support fault-tolerance and provides logarithmic search performance even for highly skewed data distributions. It supports large sets of data items for each peer. The authors experimentally tested P-Ring by using simulations as well as by using a real case of study they deployed on PlanetLab. According to the authors achievements, P-ring provide performance comparable with Skip Graphs, Online Balancing and Chord.

B. MAAN

The Multi-Attribute Addressable Network, MAAN [4] is a structured P2P system targeted to the definition of a support for resource discovery in a large distributed systems that provides support for multi-attribute range queries. It modifies the Chord hashing function by a locality preserving hashing function. In MAAN resources are identified by a set of attribute-value pairs, and each attribute is mapped on a Chord ring through the locality preserving function. The node target of the hashing function stores the full resource description so that a resource is stored as many times as its number of attributes. The resolution of a multi-attribute range query consists in executing a single 1-dimensional query on the dominant (i.e. most selective) attribute, while the other attributes are checked using the replicated data. One of the main drawbacks of MAAN is related to the large amount of memory required to store resource indices, and the cost to update them.

C. Web Retrieval P2P with high selective keys

In this contribution [9] the authors formalize an indexing and retrieval model designed to achieve high performance, cost-efficient retrieval by exploiting highly discriminative keys (HDKs) stored in a distributed global index maintained in a structured P2P network.

HDKs are carefully selected terms that appear only in a small set of collected documents. The authors present both a theoretical analysis of the scalability of the model and experimental results achieved by using the HDK-based P2P retrieval engine. These results demonstrated that the total traffic generated exploiting the HDK approach is smaller than the one obtained with distributed single-term indexing strategies. The authors also performed a deep scalability analysis that demonstrated that the HDK approach is able to scale to large networks of peers indexing very large document collections.

D. Efficient semantic search on DHT overlays

This paper [13] starts stating that distributed hash tables (DHTs) excel at exact-match lookups, but they do not directly support complex queries such as semantic search that is based on content. In this paper, the authors propose an approach to support semantic search on DHT overlays. Their idea is to place, with high probability, indexes of semantically close files into same peer nodes by exploiting information retrieval algorithms and locality sensitive hashing.

This approach adds only index information to peer nodes, causing an acceptable storage overhead. Via network simulations, the authors show that their approach is viable because the number of nodes visited for a query is about 1020, independently from the overlay size.
III. The Overall Architecture

As in [6] the system we propose (depicted in Figure 1) consists in a two-layered peer-to-peer (P2P), fully decentralized architecture. This choice is due to a series of considerations about the system goals. First of all, such systems allow to propose a service without any centralized authority (e.g., a single server that would store all user profiles), with the service implemented through the collaboration of the peers. Moreover, P2P systems allow to address the important issue of scalability, as they do not require the over- and proportional-provisioning of resources that would be required with a centralized approach. The more peers participate, the more power is added to the system. If carefully designed, a P2P approach scales up to hundred of thousands peers. Finally, P2P systems are known to deal gracefully with system dynamism at no or very reduced additional cost, whereas centralized systems need expensive and complex techniques to ensure continuous operation under node and link failures.

Starting from the peer-to-peer approach, we envision a two-layered solution. This solution aim is two-fold. Indeed, it is designed to support both a highly dynamic participation, aggregations of users and a stable and scalable maintenance of the description of the communities the users belong to.

The lower layer is in charge for the formation processes of the user communities. It has to support the construction of a interest-based network, so that peers are effectively grouped with other peers that share similar interest in their various interest domains. This requires to construct and maintain a representation of these interests (user profiling), to compare these profiles to determine their similarity (similarity metrics), to propose distributed algorithms that cluster peers in interest-based groups based on this metric (clustering algorithm) and to give a representation to the communities. Possible solutions for the constructions of this layer are beyond the scope of this paper, actually we are investigating possible solutions like [10] or [2]. The upper layer is a distributed index structure (we also refer as backbone layer). It has the responsibility to store the signature of each community as well as the contact points needed to establish a connection with that community. This allow new peers to rapidly find, among the registered communities, the best ones that match their interests. However, an index able to find only perfect matches between users and their interests could not be as effective as needed in a very dynamic and potentially heterogeneous environment. As an example consider a peer that has in common with a certain community the 90% of its characterizing items. Even if they do not share all the interests, they are probably enough similar to benefit from information exchange flows. Hence, in order to provide a effective solution, the distributed structure must be able to support the detection of the communities sharing a certain number of items with the new joining peer. To this end the index structure have to provide a proper support for similarity based searches.

IV. The Proposed Solution

As we introduced before (see Section I), a viable solution for obtaining a support to similarity searches on DHTs has been proposed in [13]. In that paper the authors applied the min-wise independent permutations approach to user profiles built according the Vector Space Model (VSM). The VSM model (also known as term vector model) is an algebraic model for representing text documents as vectors of identifiers. Documents and queries are represented as vectors of terms.

Each dimension corresponds to a separate term. If a term occurs in the document, its value in the vector is non-zero. This method is widely used due to its effectiveness in situations where a query should be compared against a given document to measure its relevance.

The main drawback of this approach is its need to have a complete knowledge on the existing terms in the considered set of documents. This is not a big problem in centralized systems that can access to the whole set of documents and, consequently, can built the whole dictionary of existing terms but is unfeasible in peer-to-peer systems where there is not a global knowledge on existing documents and terms. In order to address this problem, some complex solutions have been presented in literature [12]. However, even simpler solutions, both from the implementation and computational complexity point of view, demonstrated to achieve pretty good results [7]. One of them is the application of the
Jaccard similarity coefficient to the term vectors, that also the authors of [13] exploited in their work. Despite of its simplicity, the Jaccard Index is a very flexible measure and can be exploited to compare set of elements without specific knowledge about spatial structure of data. In order to exploit this indexing mechanism in the envisioned scenario, and pursuing the aim of finding an effective and efficient way to handle interest information, in this paper we propose three different types of profiles to model user interests. All of them are build starting from the collections of attributes that characterize a user.

The three profile types are:

- **Weighted attribute vector**: this profile consists of collections of attributes properly weighted according to a certain numerical weight indicating the attribute relevance with respect to the user profile.
- **Attribute Adjacency Matrix**: in this case the user profile is represented with a weighted attribute vector enriched with values estimating the correlation between attributes. The adjacency matrix is squared and symmetrical.
- **Set of connected components extracted from Attribute Adjacency Matrix**: the profile is represented with a set of connected component obtained by applying to the Attribute Adjacency Matrix an algorithm that extract a set of connected components. Each component describes a specific user interest.

In order to analyze the advantages deriving from the usage of all types of profiles, we performed both a theoretical and experimental comparison between two different system configurations. In the first one, we use the profiles in a DHT exploiting LSH based on the min-wise independent permutations. The second configuration consists a naive solution. As a naive solution we mean one like [4] or [3]. In those approaches, profiles are stored attribute by attribute and the query resolution process is performed by finding all the matches for every attribute and by intersecting them.

Obviously, this probabilistic method can lead to inexact query resolution but the network traffic is bounded and the storage space required is only a fraction with respect to the per attribute indexing.

V. Theoretical Analysis

Tables I, II and III show the comparison between the two indexing approaches when applied to index the user profiles respectively in the form of Weighted attribute vector, Attribute Adjacency Matrix and Set of connected components extracted from Attribute Adjacency Matrix.

In this Analysis \(|U|\) and \(|C|\) indicate respectively the number of attributes composing the profile of each user and each community. \(|A|\) represents the number of access points for each community. \(X\) indicates the number of nodes of the DHT, \(Com\) the number of existing communities. \(k\) is the maximum number of profiles requested to each node when it resolves a query. \(s\) indicates the maximum number of community profiles following from a “store” request for a new profile. \(R\) corresponds to the number of removed attributes following to a profile update.

When exploiting LSH indexing, \(n\) represents the number of identifiers associated with each profile. When the profiles are structured as Set of connected components \(|I_m|\) indicates the number of attributes describing the \(m\)-th user interest with \(\sum_{m} |I_m| \leq |U|\), whereas \(Int\) indicates the number of interests of a user.

VI. Evaluation

The validation of the proposed solution was conducted using a real dataset of users sharing a subset of the PubMed [1] scientific publications. The dataset is part of a dataset released by the Mendeley group [8]. It contains information about 5,177 users and 211,878 documents, with an average of 48.8 documents consulted by each user. The profile of each user is characterized by the terms occurring in the title, keywords and abstracts of the papers she has accessed. In order to have a more precise characterization of the users, the whole set of documents contain 127,492 distinct terms have been filtered to eliminate the stop-words, getting 127,075 “real” terms. Finally, through a process of stemming, 107,257 roots were extracted. The final set of 2,754 users exploited for the following evaluation was obtained by selecting the users with at least 150 root terms in their profiles.

Vector-based user profiles are obtained directly by these terms. Adjacency matrices are obtained by using the co-occurrences frequencies of terms within each user collection. Finally, the profiles built according the connected components model have been extracted staring from the adjacency matrices by identifying the connected components, obtained through the application of a pruning technique on the relationships occurring with a frequency less than a certain threshold \(minFreq\) and selecting, among the obtained components, those with a size greater than or equal to a certain, fixed value \(minSize\).

The following tests aim to stress the ability of the proposed system in finding a “right” community for a user that is joining the network. We used a network of 2,000 nodes, each denoting a different community representative. Users coming from the dataset are grouped in 300 different communities, obtained by using K-Means. The centroids obtained so far are used as community representatives. Moreover, the system is perturbed by the insertion of 1,700 other profiles, generated randomly in the same vector space of the user profiles.

These communities, identified using K-Means, have been used as references to compare against the ones we identified by exploiting the peer-to-peer approach. This has been performed for all three kinds of profiles.
We performed a series of tests by injecting 2,500 queries to the system. Each query is composed of a user profile. For each query, the system returns as an answer, the most similar community descriptor registered. In order to evaluate the performances of our solution, we evaluated both its efficacy and efficiency. Efficacy is measured using both network requirements for query resolution and network and storage consumption for maintaining the distributed index structure.

Figure 2. Average difference between the most similar community and the actual one

Figure 2 shows the average difference (Jaccard distance) between the most similar community found by each query and the community identified by the centralized method. As can be observed, the usage of co-occurrences data highly enhance the performance of the system. In particular, the connected components method achieves the best results. In this case, methods with more than one hashing function seems to perform better than the others. On the other hand, such achievements should be compared with the bandwidth and storage needs required to maintain indexes based on more complex profiles, like matrixes and connected components.

Figure 3. Network Bandwidth and Storage Space required by the distributed index

Figure 3 shows the global size in terms of storage space and network bandwidth, respectively, to create and to maintain the distributed index. Figure 4 depicts the global amount of network bandwidth consumption both to issue queries to the distributed index and to retrieve the results. In both the figures the results are related, respectively, to profiles built according to a vector of terms, an adjacency matrix and a set of connected components.

Figure 4. Network Bandwidth consumption for Query management

It can be noticed that, even in this case, the connected components method is the most performing one, in every kind of metrics we considered. Moreover, we compared those results with the ones obtained by a naive approach. It can be observed that every profile approach obtains a considerable gain using the LSH indexing method with respect to the pure naive solution.

VII. CONCLUSIONS

This paper has presented a DHT based approach for storing the profiles of communities of users characterized by similar interests. The similarity search over the DHT exploits the Min Wise Independent Permutation approach. We have proposed several types of user’s profiles and evaluated the indexing mechanisms with respect to the different kind of profiles. Experimental results conducted by considering a real data set have shown the effectiveness of our approach.

ACKNOWLEDGEMENT

We acknowledge the support of the S-CUBE (EU-FP7-215483), CONTRAIL (EU-FP7-257438) and RECOGNITION (EU-FP7-257756) EU projects.

REFERENCES


