A P2P REcommender system based on Gossip Overlays (PREGO)

Ranieri Baraglia  
Istituto di Scienza e Tecnologie dell’Informazione  
Consiglio Nazionale delle Ricerche  
Pisa, Italy  
Email: ranieri.baraglia@isti.cnr.it

Matteo Mordacchini  
Istituto di Scienza e Tecnologie dell’Informazione  
Consiglio Nazionale delle Ricerche  
Pisa, Italy  
Email: matteo.mordacchini@isti.cnr.it

Patrizio Dazzi  
Istituto di Scienza e Tecnologie dell’Informazione  
Consiglio Nazionale delle Ricerche  
Pisa, Italy  
Email: patrizio.dazzi@isti.cnr.it

Laura Ricci  
Dipartimento di Informatica  
Università di Pisa  
Pisa, Italy  
Email: ranieri.baraglia@isti.cnr.it

Abstract—Gossip-based Peer-to-Peer protocols proved to be very efficient for supporting dynamic and complex information exchange among distributed peers. They are useful for building and maintaining the network topology itself as well as to support a pervasive diffusion of the information injected into the network. This is very useful in a world where there is a growing need to access and be aware of many types of distributed resources like Internet pages, shared files, online products, news and information, finding flexible, scalable and efficient mechanisms addressing this topic is a key issue, even with relevant social and economic aspects. In this paper, we propose the general architecture of a system that tries to exploit the collaborative exchange of information between peers in order to build a system able to gather similar users and spread useful suggestions among them.

I. INTRODUCTION

Automating the word-of-mouth [1] is the aim of collaborative and social information filtering systems. Gossip ([2], [3]) is the term used to define a class of systems, especially some types of Peer-to-Peer (P2P) networks. Although with different intents, these two class of algorithms take inspiration from the human social behavior of spreading knowledge by exchanging information between people that are in direct contact. “Direct contact” in collaborative filtering and recommender systems means the selection of the most similar other users of a given user in order to produce recommendations of potentially interesting items. In Gossip-based P2P systems, the exchange of information between connected peers become a powerful tool to build up and maintain the network topology itself and to obtain a pervasive diffusion of the information associated with each node.

In a world where there is a growing need to access and be aware of many types of distributed resources like Internet pages, shared files, online products, news and information, finding flexible, scalable and efficient mechanisms addressing this topic is a key issue, even with relevant social and economic aspects.

Many existing proposals make use of centralized systems, like centralized search engine (e.g. Yahoo!, Google or Bing), online stores like Amazon or auction sites such as eBay. Scalability concerns are always related with these type of approaches. Another problem is that a user may need an aggregated information coming from the integration of different sources, while all these systems may provide aim only within their respective domains.

In this paper, we propose the general architecture of a system that tries to exploit the collaborative exchange of information between peers in order to build a system able to gather similar users and spread useful suggestions among them. More precisely, we wish to push further the idea of exploiting collaboratively built recommender mechanisms, based on interest clustering and obtained through interactions among users. We propose to couple the two systems by exploiting gossip-style P2P overlay networks in order to ease the gathering of users with similar interests and then use the links established so far to spread recommendations among peers. The aim that we pursue is twofold. On one hand we want to build a flexible, adaptive system that allow the creation of communities of interests among users in a decentralized, distributed way. P2P approaches (and the gossip-based ones, in particular) scale well to large numbers of peers and deal gracefully with system dynamism, whereas centralized systems need expensive and complex techniques to ensure continuous operation under node and link failures. Moreover, the service is implemented through the collaboration of the peers without needing any centralized authority that would store all profiles and ratings of users and provide centralized-controlled recommendations.

On the other hand, we want to exploit such communities not only for sharing the knowledge about interesting items inside them, but also to overcome some traditional problems of recommender systems. In particular, one of the most common problems concerns the ability to recommend new items. In the system we are proposing, each neighbor of a peer $P$ will
push recommendations to it about items that it believes might be of potential interest for \( P \). This decision is taken locally, when a neighbor selects or knows from its connections of the existence of a new item whose characteristics are related with one or more of its communities. It can then suggest this item to \( P \) and its other neighbors of all the related communities. This mechanism would allow a more efficient and rapid diffusion of the information.

The remain of this paper is organized as follows: in Sec. II we revise the literature about the subject of this paper; in Sec. III we present the architecture of the proposed system, while in Sec. IV we give an experimental evaluation of our approach. In Sec. V, conclusions and possible further exploitations of this work are examined.

II. RELATED WORK

Many independent studies have observed the characteristics of accesses to distributed data in various contexts. Among them, in the context of this paper, we focus on: the clustering of the graph that links users based on their shared interests, the correlation between past and futures accesses by users (or by groups of users) that share similar interests, the skewness of the distribution of interests per peer, the skewness of the distribution of accesses per data element. Skewness usually relates to Zipf-shape distributions, which are a feature of access behaviors amongst large groups of humans [4]. We first review the work related to the detection and use of interest correlation between users in large-scale systems. The presence of communities amongst user interests and accesses in Web search traces [5]–[7], peer-to-peer file sharing systems [8] or RSS news feeds subscriptions [9] can be exhibited. The existence of a correlation of interests amongst a group of distributed users has been leveraged in a variety of contexts and for designing or enhancing various distributed systems. For peer-to-peer file sharing systems that include file search facilities (e.g., Gnutella, eMule, ...), a sound approach to increase recall and precision of the search is to group users based on their past search history or based on their current cache content [10]–[12]. Interestingly, the small-world [13] aspects of the graph of shared interests linking users with similar profiles is observed and can be exploited not only for file sharing systems, but also in researcher communities or in web access patterns [8]. Another potential use of interest clustering is to form groups of peers that are likely to be interested in the same content in the future, hence forming groups of subscribers in a content-based publish-subscribe system [14], [15]. Moreover, interest correlation can be used to help bootstrapping and self-organization of dissemination structures such as network-delay-aware trees for RSS dissemination [16]. Finally, user interest correlation can be used for efficiently prefetching data in environments where access delays and resource usage constraints can be competing [17], as it is an effective way of predicting future accesses of the users with good accuracy.

The correlation between the users past and present accesses has been used for user-centric ranking. In order to improve the personalization of search results, the most probable expectations of users are determined using their search histories stored on a centralized server [18], [19]. Nevertheless, the correlation between users with similar search histories is not leveraged to improve the quality of result personalization, hence making the approach sound only for users with sufficiently long search histories. An alternative class of clustering search engines uses semantic information in order to cluster results according to the general domain they belong in (and not as in our approach to cluster users based on their interests). This can be seen as a centralized, server-side and user-agnostic approach to the use of characteristics of distributed accesses to improve user experience. The clustering amongst data elements is derived from their vocabulary. It presents the user with results along different interest domains and can help her to disambiguate these results from a query that may cover several domains, e.g., the query word apple can relate to both food/fruits and computers domains. Examples of such systems are EigenCluster [20], Grouper [21], SnakeT [22] or TermRank [23]. Nonetheless, these systems simply modify the presentation of results so that the user decides herself in which domain the interesting results may fall; these results are not in any way automatically tailored to her expectations. They do not also consider the clustering of interest amongst users, but only the clustering in content amongst the data. Other approaches cluster users on the basis of similarity between their semantics profile. Approaches of this kind of systems includes GridVine [24], the semantic overlay networks [25] and p2pDating [26]. They build a semantic overlay infrastructure based on a peer-to-peer access structure. It relies on a logical layer storing data. In order to create links among peers they use schemas, and schema mappings. They make use of heterogeneous but semantically related information sources whereas our approach does not rely on any kind semantic interpretation. It, in principle, enables a broader exploitation of more heterogeneous data sources.

III. PROPOSED APPROACH

In this section we present the general principles for the construction and use of a network of peers based on shared interest. Links between peers are established when they have been interested in the same content in the past. Commonly, they are considered to potentially show interests for the same content in the future. Thus, they can collaboratively exchange useful recommendations among them.

In order to group similar users, the protocol works by the means of a clustering algorithm. First, each peer decides, independently, which are the peers it is linked with. These

---

1 Small-world aspects for the shared interests graph are: (i) a high clustering, (ii) a low diameter due to the existence of a small proportion of long links, i.e., links to exotic domains that are distant from the common interests of the node and its regular neighbors and that act as cross-interest-domain links, and (iii) the possibility to navigate through the graph of interest proximity amongst peers and effectively find short path between two interest domains based only on the one-to-one distance relationships amongst these domains, i.e., without global knowledge of the graph.
one-to-one relationships are chosen on the basis of an interest-based distance, amongst the peer it encounters. Every time it get in touch with a new peer, it can learn of the existence of new potential neighbors (i.e., similar users) and then communicate with them. Finally, it can learn of other, potentially better neighbors. When this process is stabilized, a peer may consider its neighborhood as the the representative of a community of a shared interest. An important point here is that a peer is characterized by multiple interests. Hence, the process depicted above is conducted separately for each of the interests of a user. Connections are established and maintained separately for every distinct interest. Thus, we have a set of virtual different overlays, where each peer participate in as many groups it requires to cover its interests. In order to achieve this organization, each peer start a different stream of messages for each of its distinct interests.

A. Users Profile

We want to have profiles of peers created using the users’ interests. They can be represented by recently accessed resources, purchased items, visited pages, etc. Such gathered information have to be considered in a proper way, since it forms the basis over which the whole network will be built up. Generally speaking, let \( \Theta \) be the set of items belonging to a peer \( p \). We consider that the profile \( \pi \) of \( p \) can be defined as

\[
\pi_p = \{(i, C(i), R(i)), \forall i \in \Theta \}
\]

where \( i \) is an item of the set of \( \Theta \), \( C(i) \) is the content associated with \( i \) (potentially void) and \( R(i) \) is the rating given by \( p \) to \( i \). Moreover, \( p \) has a set \( I = \{I_1, \ldots, I_k\} \) of interests. Each of the items in \( \pi_p \) may be associated with an interest \( I_j \).

We can then represent \( \pi_p \) in the following way:

\[
\pi_p = \bigcup_{j=1,\ldots,k} \pi_p(I_j)
\]

where \( \pi_p(I_j) \) is the set of items associated with the interest \( I_j \).

Having a suitable method to describe each user interests coded in the peer profiles, we have to put attention on choosing a proper function similarity function \( \text{sim} : \Pi^2 \rightarrow \mathbb{R} \) to compare profiles, where \( \Pi \) is the set of all the profiles. This is a particular relevant point, since this function determines the relationships between peers on the basis of their interests. If each different interest is determined by different type of features, different similarity measures could be used to evaluate peers proximities with respect to each interest.

Several measures can be exploited to this end. For instance, one common way is to use a metric that takes into account the size of each profile, such as the Jaccard similarity,

\[
\text{sim}(p_1, p_2) = \frac{|\pi_{p_1} \cap \pi_{p_2}|}{|\pi_{p_1} \cup \pi_{p_2}|},
\]

that have proven to be an effective similarity measure [11], [16].

B. Creation of Interest Communities

As soon as each peer is able to compute its interest-based distance to any other peer, its objective is to group with other peers that have close-by interests, in order to form the basis for interests communities. This process is done in a self-organizing and completely decentralized manner, using a gossip-style communication. Each peer knows a set of other peers, namely its interest neighbors, and tries periodically to choose new such neighbors that are closer to its interest than the previous ones. This is simply done by learning about new peers from some other peer, then retrieving their profile, and finally choosing the \( C \) nearest neighbors in the union of present and potential neighbors.

When a peer \( p \) enters the network, it is put in contact with one or more peers already taking part in the interest-proximity network. They use the profile similarity function to compute how similar they are. They consider each interest in the \( I \) of \( \pi_p \) separately and they confront it with their own. Moreover, the peers contacted by \( p \) use the same similarity function to determine which are, among their neighbors, the most similar to \( p \) and route the join request of \( p \) toward them. All the peers that receive that request will react using the same protocol described above. All the interactions are shown in Algorithms 1 and 2. This mechanism will lead \( p \) to learn the existence of the most similar nodes in the network and allow it to connect with them. In doing this process, the involved peers can only use their local knowledge to compare their respective profiles.

**Algorithm 1** A peer \( P \) gossiping active process

Let \( N(P) \) be the set of \( P \)'s actual neighbors

for all \( P_i \in N(P) \) do

Get from \( P_i \) a set \( \text{NewPeers} \) from its neighborhood

for all \( P' \in \text{NewPeers} \) do

if \( P' \notin N(P) \) then

connect with \( P' \)

if \( \text{Sim}(P, P') \geq \min_{P_j \in N(P)} \text{Sim}(P, P_j) \) then

add \( P' \) to \( N(P) \)

end if

end if

end for

end for

Once the process is stabilized, \( p \) can consider its neighbors as the representatives of a personal community of “friends” from which request and to which forward recommendations. Thus, the gossip protocol provide the basis for classical recommender systems in forming the set of similar users. This is done distributely and adaptively and the gossip protocol ensure a robust and constant maintenance over time. Recommendations can then be requested by \( p \) to its neighborhood and it can forward the newly items it discovered to its neighbors.

IV. Preliminary results

In order to evaluate the envisioned architecture and the algorithms presented in this paper, with respect to their ability to build a peer-to-peer system able to group users sharing
Algorithm 2 A peer $P$ gossiping passive process

Let $CR(P')$ be a connection request from another peer $P'$. Let $NewPeers = \emptyset$. If $\text{Sim}(P, P') \geq \min_{P_i \in N(P)} \text{Sim}(P, P_i)$ then accept $CR(P')$. For all $P_i \in N(P)$ do:

- If $\text{Sim}(P_i, P') \geq \text{threshold}$ then add $P_i$ to $NewPeers$.

End if.

End for.

Add $P'$ to $N(P)$. Send $NewPeers$ to $P'$.

Else refuse $CR(P')$.

End if.

Algorithm 3 Recommendation response process

Let $N_P(I_j)$ be the set of $P'$s neighbors for the interest $I_j$. Receive a recommendation request from $p' \in N_P(I_j)$. For all $i \in \pi_P(I_j)$ do:

- If $\text{Sim}(p', i) \geq \text{threshold}$ then recommend $i$ to $p'$.

End if.

End for.

Algorithm 4 Recommendation suggestion

Know about a new item $h$. Let $I_j$ be the interest $h$ is related to. Let $N_P(I_j)$ be the neighborhood of peers interested in $I_j$. For all $p' \in N_P(I_j)$ do:

- If $\text{Sim}(p', h) \geq \text{threshold}$ then recommend $h$ to $p'$.

End if.

End for.

common interests in a totally decentralized way, we developed a prototype implementing the gossip-based peer-to-peer protocols described by the Algorithms 1, 2.

The prototype has been developed and tested using the Overlay Weaver peer-to-peer framework. Overlay Weaver is a framework implementing various P2P overlays which aims at separating high level services such as DHT, multicast and anycast from the underlying key-based routing (KBR) level. Its architecture is depicted in Fig. 2. The routing layer architecture follows the KBR concepts but leaves behind the KBR monolithic approach, decomposing the routing layer in a set of independent modules, (e.g. communications, routing and query algorithms). The routing module is defined by three layers: the routing layer (bottom), the service layer and the application layer (top). The main advantages coming from the usage of that simulator are the rapid prototyping (with a simulator it is not required to deal with low-level distributed development issues, e.g. network socket programming, fault management), the possibility of simulating the behavior of the system in a big network (even composed of thousands of nodes) without having that network, and the possibility of define a network observer able to monitor the status of the network in any time.

In the current prototype the user profiles (see Figure 1) contain data about the peer neighborhood, the neighbors profiles and the middle-range frequently sites visited by user. For each site visited by the user and belonging to MRFVS set the profile stores the URL and the number of visits.

During the first testbed the prototype has been tested by
using as input an automatically-generated synthetic dataset. That dataset defines the profiles of ten thousand users, each user profile holds a variable number of MRFVS URLs ranging from five to twenty. Using this dataset we performed the several experiments, varing both:

- the function used by the peer for selecting the peers with which to communicate and to exchange data;
- the amount of data exchanged from peer to peer in terms of the maximum number of profiles the sender peer transfers to the receiver peer.

The results obtained by this testbed are represented in Figure 5. The Figure is divided in two parts, the left part of the figure shows the results achieved using a random peer selection function whereas the right part depicts the results obtained when the selection function chooses the most similar peers among the ones in the peer neighborhood.

On the x-axis are indicated the simulator cycle numbers, roughly speaking it can be described as the number of times the peers computed their gossiping algorithm. This measure is useful for evaluating the convergence speed of the algorithm, namely the amount of time (or cycles) required by the algorithm to find a good set of neighbors. In our case “good” means similar. On the y-axis is reported the average similarity degree of the peer profile with respect to the profiles of its neighborhood.

It is easy to see that the solution based on the similarity-driven function for neighbors selection produces significantly better results even when the data exchanged for each iteration is low.

The aim of the second test we conducted was the scalability of the prototype with respect to the number of peers in the network. In this case the function used for selecting the peers to communicate with is the similarity-driven only. The shuffle length has been fixed to ten, which demonstrated in the first testbed to be a good tradeoff between the quality of the built neighborhood and the amount of data exchanged.

We can observe that when there are too few peers in the network, the algorithm performance gets worse. Instead, with at least 1,000 peers, we obtain almost the same results, with a high degree of similarity among the computed neighborhood.

In order to have a further evaluation of the proposed approach, we run a series of tests using the publicly available, well-known and widely-used Movielens dataset. In this case,
The results are shown in Fig. 7. we used 10,000 users, choosing at most 20 movies per user. The results are shown in Fig. 7.

Note that even with this dataset the proposed algorithm gives a great boost compared with a traditional, gossip-based algorithm, as for the synthetic data. Moreover, it builds very compact neighborhoods (with respect to the used similarity metric), thus giving a strong representation of the community a peer might belong to.

V. CONCLUSION

The focus of this paper is on addressing the problem of clustering users in a purely decentralized way. This is particularly useful for enabling an automated creation of communities made from users sharing common interests. In this paper we presented the overall architecture of a gossip-based peer-to-peer system exploiting collaboratively built search mechanisms.

ACKNOWLEDGMENT

The authors acknowledge the support of Project FP6-033576, Building and Promoting a Linux-based Operating System to Support Virtual Organizations for Next Generation Grids (2006-2010), and the CNR program “Ricerca a tema libero”. The authors would like also to thank Matteo Fulgeri, who conducted several experiments.

REFERENCES


