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: ricci@di.unipi.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 aid 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

978-0-7695-4108-2/10 $26.00 © 2010 IEEE
DOI 10.1109/CIT.2010.502

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 od 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], [6], peer-to-peer file sharing systems [7] or RSS news feeds subscriptions [8] 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 [9]–[11]. Interestingly, the small-world [12] aspects of the graph of shared interests[1] 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 [7]. 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 [13]. Moreover, interest correlation can be used to help bootstrapping and self-organization of dissemination structures such as network-delay-aware trees for RSS dissemination [14]. Finally, user interest correlation can be used for efficiently prefetching data in environments where access delays and resource usage constraints can be competing [15], 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 [16], [17]. 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 the user 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 [18], Grouper [19], SnakeT [20] or TermRank [21]. 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 [22], the semantic overlay networks [23] and p2pDating [24]. 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.

Related to out proposal is Tribler [25], a P2P television recommender system. It is a system able to recommend, record and download television programs to/from its users. It construct a social P2P network. Anyway, in contrast with our approach, neighbor lists (called *buddy tables*) can be directly filled in by the user herself using an interface. No topology or affinity property is considered. We propose a gossip system that construct and maintain in rest groups of dynamic users based on their past activities, without needing their direct intervention.

[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.

III. Proposed approach

In this section we present the general principles for the construction and use of a network of peers based on shared interests. We use these principles as the basis of our PREGO network, a P2P system that exploits multiple gossip overlays to spread recommendations among peers. A representation of a network of this kind is given in Fig.1. Links between peers are established when they have been interested in the same content in the past. In general, they are considered to potentially show interests for the same content in the future. Thus, peers collaboratively exchange useful recommendations among themselves. 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 one-to-one relationships are chosen on the basis of an interest-based distance, amongst the peer it encounters. Every time it gets 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 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 peer. Connections are established and maintained separately for every distinct interest. Thus, we have a set of virtual different overlays, where each peer participates in as many groups is required to cover its interests. The situation is showed in Fig.2. In order to achieve this organization, each peer start a different stream of messages for each of its distinct interests.

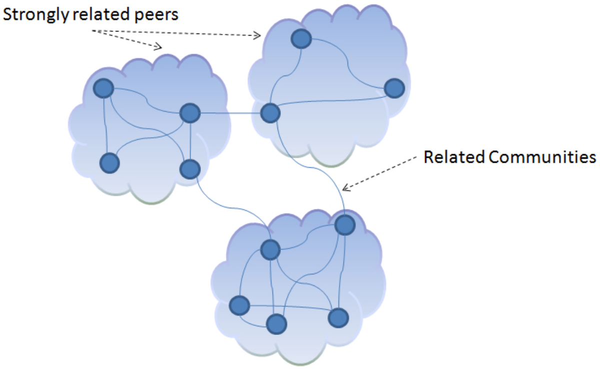


Fig. 1. Interest Communities

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 has to be considered in a proper way, since it forms the basis over which the whole network will be built up. Generally speaking, let $\Im$ be the set of items of the system and let $\Im_p \subseteq \Im$ be the subset of items belonging to a peer p. We consider that the profile π of p can be defined as

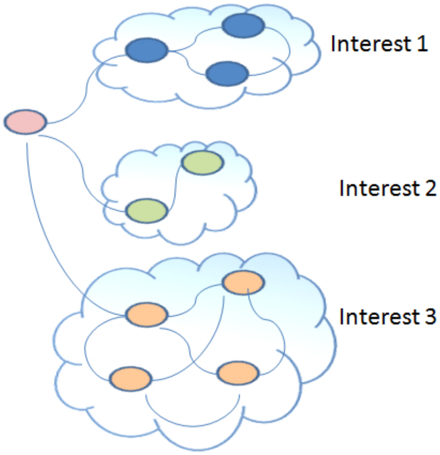


Fig. 2. Interest Overlays

$$\pi_p = \{(i, C(i), R(i)) | i \in \Im_p\}$$

where i is an item of the set of $\Im_p$, $C(i)$ is the content associated with i (potentially void) and $R(i)$ is the rating given by p to i. Moreover, p has a set $I^p = \{I^p_1, \ldots, I^p_k\}$ of interests. Each of the items in π_p may be associated with an interest I^p_j. We can then represent π_p in the following way:

$$\pi_p = \bigcup_{j=1,\ldots,k} \pi_p(I^p_j)$$

where $\pi_p(I^p_j)$ is the set of items associated with the interest I^p_j.

For performing this association, we assume each peer has a function γ that given an item belonging to $\Im$ decides the interest it should be associated with. More formally:

$$\gamma_p(i) = I^p_j \quad with\ i \in \Im_p$$

Note that the set I^p is specific for each distinct peer p. We do not assume any globally known labeling, categorization or subdivision of the objects in $\Im$. Each peer organizes its own subdivision of $\Im_p$ in the interests of I_p. It can then confront its objects divided per interest with the sets of the other peers it will be put in contact. Given two peers p_1 and p_2, p_1 will consider its local interest $I^{p_1}_s$ similar to the interest $I^{p_2}_t$ if it contains the most similar set of items among the other sets in I^{p_2} with respect to the items in $I^{p_1}_s$.

Thus, 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 $sim : \Pi^2 \to \mathbb{R}$ to compare profiles, where Π is the set of all possible 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, that has proven to be an effective similarity measure [10], [14]. Given two peers p_1 and p_2 and two interests $I_s^{p_1}$ and $I_t^{p_2}$, the similarity can be computed as

$$sim(p_1, p_2) = \frac{|\pi_{p_1}(I_s^{p_1}) \cap \pi_{p_2}(I_t^{p_2})|}{|\pi_{p_1}(I_s^{p_1}) \cup \pi_{p_2}(I_t^{p_2})|}$$

One possible use of this measure is shown in Fig. 3. In this figure, we assume that $\Im_p$ is composed of visited URLs and $R(i)$ consists of the visiting frequency of each URL.

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 π_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 gossip active process for each interest

```
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 $NewPeers$ from its neighborhood
    for all $P^{'} \in NewPeers$ do
        if $P^{'} \notin N(P)$ then
            connect with $P^{'}$
            if $Sim(P, P^{'}) \geq \min_{P_j \in N(P)} Sim(P, P_j)$ then
                add $P^{'}$ to $N(P)$
            end if
        end if
    end for
end for
```

Algorithm 2 A peer P gossip passive process for each interest

```
Let $CR(P')$ be a connection request from another peer $P'$
Let $NewPeers = \emptyset$
if $Sim(P, P^{'}) \geq \min_{P_i \in N(P)} Sim(P, P_i)$ then
    Accept $CR(P')$
    for all $P_i \in N(P)$ do
        if $Sim(P_i, P') \geq \theta$ 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
```

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 distributively 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 using Algorithms 3 and 4.

Algorithm 3 Recommendation response process

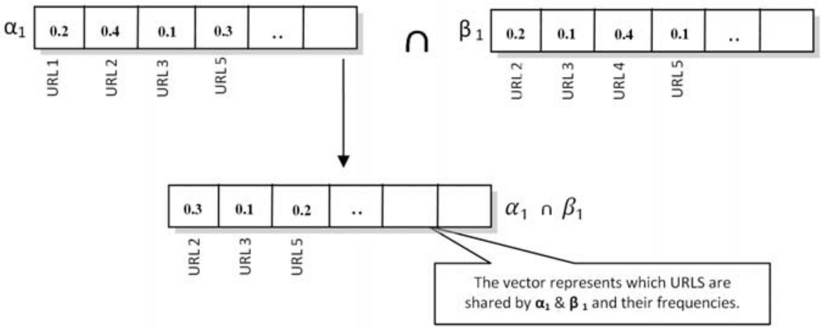
```
Let $N_p(I_j)$ be the set of $P'$s neighbors for the interest $I_j$
Receive a recommendation request from $p' \in N_P(I_j)$
for all $i \in \pi_p(I_j)$ do
    if $Sim(p', i) \geq \theta$ then
        recommend $i$ to $p'$
    end if
end for
```

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 $Sim(p', h) \geq \theta$ then
        recommend $h$ to $p'$
    end if
end for
```

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 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 vector represents which URLS are shared by α_1 & β_1 and their frequencies.

Fig. 3. User Profile

The prototype has been developed and tested using the Overlay Weaver [26] 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. 4, that is the official architecture picture published in the Overlay Weaver Web Site. 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.

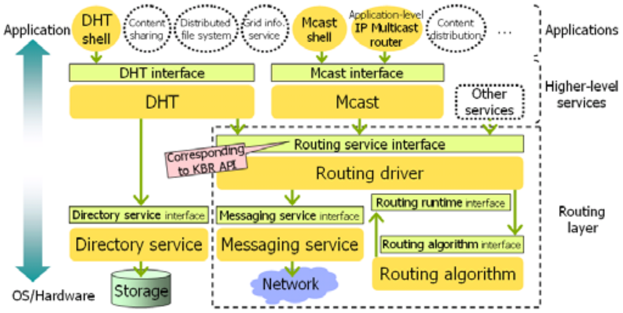


Fig. 4. Overlay Weaver Architecture [26]

The data used for the experiments comes from the Movielens Dataset [27]. This is a well-known dataset for testing collaborative filtering algorithms. The dataset is a movie recommendation dataset created by the Grouplens Research Project. The data we used consists of 1 million ratings for 3900 movies by 6040 users. Since each user only rates a few movies, the data matrix is very sparse. This makes this dataset a good benchmark for our approach because the construction of communities with a sparse dataset is a real hard task. The choice of using the Movielens dataset affected the structure of user profiles and their representation. In the current prototype each user profile contains information about the movies seen by the respective user. Each peer stores the information about its neighborhood: the addresses and the profiles of its neighbors.

Using this dataset we performed the several experiments varying:

1) the functions used for selecting both the peers with which to communicate and the best peers to send among the ones in the neighborhood;
2) the neighborhood size;
3) the number of peers in the network;
4) the number of gossip-protocol iterations performed before measuring the similarity among a peer and its neighborhood;

Regarding the points 1-2, we implemented three different gossip protocols in the simulator: Cyclon, Vicinity [28] and Twinfinder. All of them have a behavior compliant with the Algorithms 1 and 2. Cyclon and Vicinity [28] are well known Algorithms, whereas Twinfinder is a customized version of Vicinity, we conceived and designed, where the sender transfer to each neighbor a potentially different set of profiles, in particular the ones considered the most similar to the receiver. Each protocol has been tested under several different conditions, varying the maximum number of neighbors a peer can store (in the range [1-20]) and the number of nodes in the networks (during this first experimental phase in the range [1000-3000]). The results we achieved are depicted in Figure 5, this figure is in turn composed of eight subfigures organized in four rows and two columns. In the first three rows, each row represent a different protocol. The subfigures in the first column shows the average similarity, computed using the Jaccard similarity metrics, between a peer and its neighborhood. The second column reports the variance and the standard deviation of results obtained by each peer in the network for each protocol implemented. Finally, the last row shows the comparison among the results provided by the three

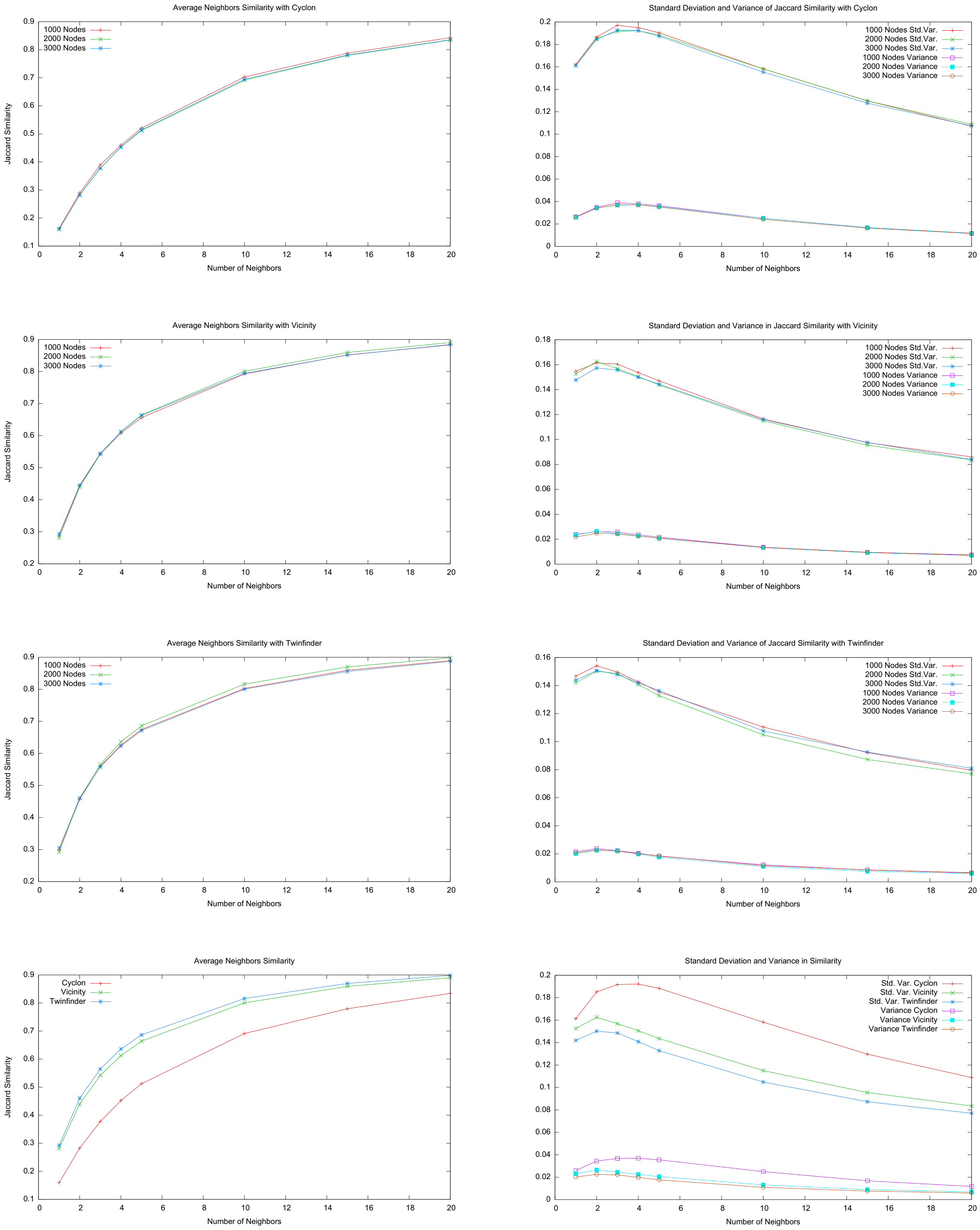


Fig. 5. Average Jaccard Similarity, Variance and Standard Deviation with Cyclon, Vicinity, Twinfinder

protocols when the network is made of 2000 peers, namely the half-way in the range [1000-3000]. It is easy to see that the two protocols, Vicinity and Twinfinder, exploiting the peer profile similarity in the peer selection process achieve better results. Twinfinder, in particular, obtained a sensible better values in terms of standard deviation.

Those experiments gave us a further confirmation that the gossip protocols, and our Twinfinder in particular, are suitable solutions to build links among similar peers, hence for building up Interest Communities. Then, we decided to perform experiments for measuring the potential ability of gossip protocols for providing recommendations inside the Interest Communities. In order to do it we defined the “Coverage” measure. Given the set of movie genres liked by a user, we defined her Coverage as the percentage of user genres potentially recommendable to her by its neighborhood. In this case, we used as a simple function γ_p the distinction between genres of the movies in the dataset. Figure 6 shows in the two subfigures contained, respectively, the Coverage achieved by the three protocols and the number of peer interactions with which that coverage is obtained. Also in this case the similarity based protocols, and Twinfinder in particular, achieved better results both in terms of Coverage and in terms of Coverage speed. The algorithm convergence speed measurement is useful for evaluating the amount of time (or cycles) required by the gossip protocol to find a good set of neighbors.

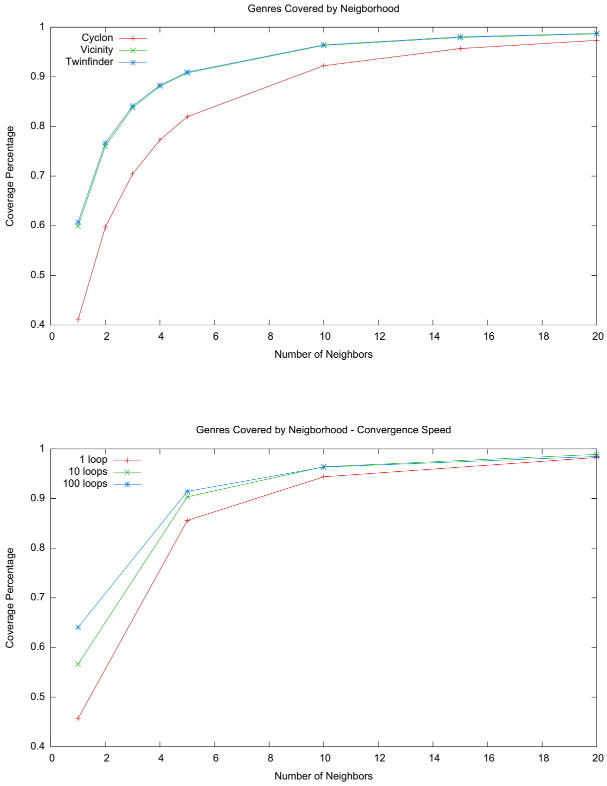


Fig. 6. Neighborhood Coverage Percentage and Speed

The last testbed regarded the ability of the approach we present to provide good results either when the network is small or when it is big. In this case we measured the Coverage when the network size is 1000, 2000, 3000 and 5000. The experiment shows best results when the network size is big, this is a further confirmation of gossip protocols ability to build up links with similar peers, indeed, if on one side an increased network size, fixed a certain user, raises, the probability to find a user in the network close to her, on the other side lowers the probability, given two peers, they keep in contact by accident.

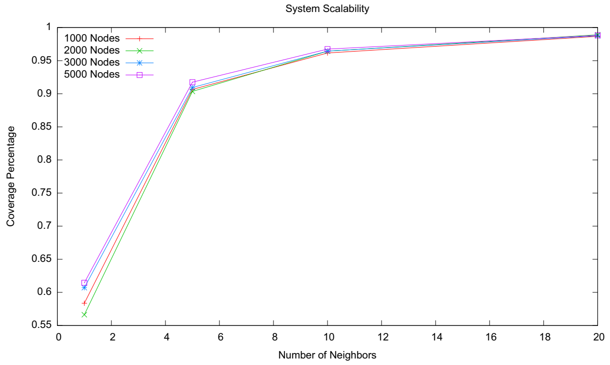


Fig. 7. Scalability

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

[1] U. Shardanand and P. Maes, “Social information filtering: algorithms for automating “word of mouth”,” in *Proc. of the SIGCHI conference on Human factors in computing systems*, ACM. New York, NY, USA: Addison-Wesley, 1995, pp. 210–217.

[2] M. Jelasity, R. Guerraoui, A.-M. Kermarrec, and M. van Steen, “The peer sampling service: experimental evaluation of unstructured gossip-based implementations,” in *Middleware ’04: Proceedings of the 5th ACM/IFIP/USENIX international conference on Middleware*. New York, NY, USA: Springer-Verlag New York, Inc., 2004, pp. 79–98.

[3] M. Jelasity, S. Voulgaris, R. Guerraoui, A.-M. Kermarrec, and M. van Steen, “Gossip-based peer sampling,” *ACM Trans. Comput. Syst.*, vol. 25, no. 3, p. 8, 2007.

[4] G. K. Zipf, *Human Behavior and the Principle of Least Effort*. Harvard University Press, 1949.

[5] R. Akavipat, L.-S. Wu, F. Menczer, and A. Maguitman, “Emerging semantic communities in peer web search,” in *Proc. of P2PIR’06*, 2006, pp. 1–8.

[6] L. A. Adamic and B. A. Huberman, “Zipf’s law and the internet,” *Glottometrics*, vol. 3, pp. 143–150, 2002.

[7] A. Iamnitchi, M. Ripeanu, and I. Foster, “Small-world file-sharing communities,” *ArXiv Computer Science e-prints*, Jul. 2003.

[8] H. Liu, V. Ramasubramanian, and E. G. Sirer, “Client behavior and feed characteristics of RSS, a publish-subscribe system for web micronews,” in *Proceedings of the 2005 Internet Measurement Conference (IMC 2005)*, Oct. 2005.

[9] K. Sripanidkulchai, B. Maggs, and H. Zhang, “Efficient content location using interest-based locality in peer-to-peer systems,” in *IEEE Infocom, San Francisco, CA, USA*, Mar. 2003. [Online]. Available: http://www.ieee-infocom.org/2003/papers/53_01.PDF

[10] P. Fraigniaud, P. Gauron, and M. Latapy, “Combining the use of clustering and scale-free nature of user exchanges into a simple and efficient p2p system.” in *Proc. of EuroPar’05*, 2005.

[11] S. B. Handurukande, A.-M. Kermarrec, F. Le Fessant, L. Massoulié, and S. Patarin, “Peer sharing behaviour in the edonkey network, and implications for the design of server-less file sharing systems,” in *Proceedings of Eurosys’06*, Leuven, Belgium, apr 2006.

[12] J. Kleinberg, “Navigation in a small world,” *Nature*, vol. 406, 2000.

[13] R. Chand and P. A. Felber, “Semantic peer-to-peer overlays for publish/subscribe networks,” in *Proceedings of Europar’05, European Conference on Parallel Processing*, Lisboa, Portugal, Sep. 2005, pp. 1194–1204.

[14] J. A. Patel, E. Rivière, I. Gupta, and A.-M. Kermarrec, “Rappel: Exploiting interest and network locality to improve fairness in publish-subscribe systems,” *Computer Networks*, 2009, in Press.

[15] T. Wu, S. Ahuja, and S. Dixit, “Efficient mobile content delivery by exploiting user interest correlation,” in *Proc. WWW (Poster)*, 2003.

[16] B. Tan, X. Shen, and C. Zhai, “Mining long-term search history to improve search accuracy,” in *Proc. of SIGKDD’06*, Philadelphia, PA, USA, 2006, pp. 718–723. [Online]. Available: http://portal.acm.org/citation.cfm?id=1150493

[17] J. Teevan, S. T. Dumais, and E. Horvitz, “Personalizing search via automated analysis of interests and activities,” in *Proc. of SIGIR-IR’05*, Salvador, Brazil, 2005, pp. 449–456. [Online]. Available: http://portal.acm.org/citation.cfm?id=1076111

[18] D. Cheng, R. Kannan, S. Vempala, and G. Wang, “A divide-and-merge methodology for clustering,” *ACM Trans. Database Syst.*, vol. 31, no. 4, pp. 1499–1525, 2006.

[19] O. Zamir and O. Etzioni, “Grouper: a dynamic clustering interface to web search results,” *Computer Networks*, vol. 31, no. 11-16, pp. 1361–1374, 1999.

[20] P. Ferragina and A. Gulli, “A personalized search engine based on web-snippet hierarchical clustering,” *Softw. Pract. Exper.*, vol. 38, no. 2, pp. 189–225, 2008.

[21] F. Gelgi and H. D. S. Vadrevu, “Term ranking for clustering web search results,” in *Proceedings of the 10th International Workshop on Web and Databases (WebCD 2007)*, Beijing, China, jun 2007.

[22] K. Aberer, P. Cudre-Mauroux, M. Hauswirth, and T. V. Pelt, “Gridvine: Building internet-scale semantic overlay networks,” in *In International Semantic Web Conference*, 2004, pp. 107–121.

[23] A. Crespo and H. Garcia-Molina, “Semantic overlay networks for p2p systems,” Stanford University, Technical report, Computer Science Department, 2002. [Online]. Available: http://citeseer.ist.psu.edu/568934.html

[24] J. X. Parreira, S. Michel, and G. Weikum, “p2pdating: Real life inspired semantic overlay networks for web search,” *Inf. Process. Manage.*, vol. 43, no. 3, pp. 643–664, 2007.

[25] J. A. Pouwelse, P. Garbacki, A. B. J. Wang, J. Yang, A. Iosup, D. H. J. Epema, M. Reinders, M. R. van Steen, and H. J. Sips, “Tribler: a social-based peer-to-peer system,” *Concurrency and Computation: Practice and Experience*, vol. 20, no. 2, pp. 127–138, 2008.

[26] S.Kazuyuki, T.Yoshio, and S.Satoshi, “Overlay Weaver: An Overlay Construction Toolkit,” *Computer Communications*, vol. 31, no. 2, pp. 402–412, February 2008.

[27] B. N. Miller, I. Albert, S. K. Lam, J. A. Konstan, and J. Riedl, “Movielens unplugged: experiences with an occasionally connected recommender system,” in *IUI ’03: Proceedings of the 8th international conference on Intelligent user interfaces*. New York, NY, USA: ACM, 2003, pp. 263–266.

[28] S. Voulgaris and M. van Steen, “Epidemic-style management of semantic overlays for content-based searching,” in *Euro-Par*, ser. Lecture Notes in Computer Science, J. C. Cunha and P. D. Medeiros, Eds., vol. 3648. Springer, 2005, pp. 1143–1152.