Clustering in Peer-to-Peer File Sharing Workloads

F. Le Fessant†, S. Handurukande‡, A.-M. Kermarrec⋆ & L. Massoulié⋆
†INRIA-Futurs and LIX, Palaiseau, France
‡EPFL, Lausanne, Switzerland
⋆ Microsoft Research, Cambridge, UK

Abstract

Peer-to-peer file sharing systems now generate a significant portion of Internet traffic. A good understanding of their workloads is crucial in order to improve their scalability, robustness and performance. Previous measurement studies on Kazaa and Gnutella were based on monitoring peer requests, and mostly concerned with peer and file availability and network traffic. In this paper, we take different measurements: instead of passively recording requests, we actively probe peers to get their cache contents information. This provides us with a map of contents, that we use to evaluate the degree of clustering in the system [1], and that could be exploited to improve significantly the search process.

1 Introduction

File sharing using peer-to-peer (P2P) networks has gained wide popularity; some reports [8, 10] suggest that P2P traffic is the dominant consumer of bandwidth ahead of Web traffic. This popularity triggered a lot of research activity. While one research trend aims at improving the performance and features [9, 14, 3], another trend is concerned with analysing and modeling these networks.

In the last few years, many measurement analysis of P2P networks focusing on *Free-Riding* [1], peer connectivity and availability (e.g., [2, 12, 11]), peer distribution and locality within the network (in [12, 6]) were carried out. But to the best of our knowledge no detailed analysis has been done on the type of content shared on these networks.

Recent studies [6] show that locality-awareness in P2P networks can improve their efficiency significantly: by clustering users according to their *geographical* locality, file requests can be answered faster.

Similarly, clustering based on the types of files that peers have, also called *interest-based* locality, may improve search performance: it may reduce the duration of the search phase in general, and more specifically when searching for rare files [13]. Obviously, the performance gains obtained from exploiting either type of locality will depend on the corresponding degree of clustering. Our work investigates these issues.

Contribution. In this paper we present an analysis of contents (the type of files: music, video, documents etc) that peers offer to others in P2P file sharing networks. We used an active probing technique which allowed to capture the lists of files that peers offer. The community that we probed used eDonkey 2000, Kazaa's main competitor among all the P2P networks [7]. We then examine the clustering properties of the observed workload. Our preliminary findings show that both geographical and interest-based clustering are present and could thus be leveraged in order to yield significant performance improvements.

2 Experimental Setup

2.1 The eDonkey Network

The eDonkey 2000 network is one of the most advanced P2P file-sharing network, providing features such as concurrent download of a file from multiples sources, detection of file corruption using hashing, partial sharing of files while being

[1] We define clustering here as the overlap between cache contents.

downloaded and expressive querying methods for file search. According to a recent study [15], in some European countries eDonkey is now ahead of Kazaa in terms of resulting traffic.

The eDonkey network is a hybrid two-layer network composed of clients and servers. The clients connect to a server and register the files that they are sharing, providing the meta-data describing the files. Once registered, the clients can either search by querying meta-data or request a particular file via its unique network identifier. Servers provide the locations of files when requested by clients, so that clients can download the files directly from the provided locations.

2.2 Methodology

Measurements reported in this paper have been obtained by crawling the eDonkey network during the first week of November 2003. 230,000 eDonkey clients were discovered; 55,000 of them were connected for 3 days. We kept 37,000 clients that could clearly be identified as distinct clients, among which 25,000 clients shared no files at all. Our study is based on the 923,000 different files shared by the remaining 12,000 clients.

Our crawler has been implemented by modifying an existing eDonkey client, namely MLdonkey [5]. Our modified client runs two concurrent tasks: discovering eDonkey clients and scanning their contents.

Client Discovery. Our crawler first connects to as many eDonkey servers as possible requesting their list of clients (server responses are limited to 200 clients). In order to obtain as many clients as possible from each server (servers may have up to 100,000 simultaneously connected users), we send requests for clients containing the strings "aa", "ab", "ac", ... and so on until "zz".

Client Content Scanning. Our crawler attempts to connect to every eDonkey client that is discovered. If it succeeds, it obtains the unique identifier of the client and requests its list of shared files. All these data are then stored in a file, named using a digest of the IP address and port of the client, so that the resulting trace is completely anonymised.

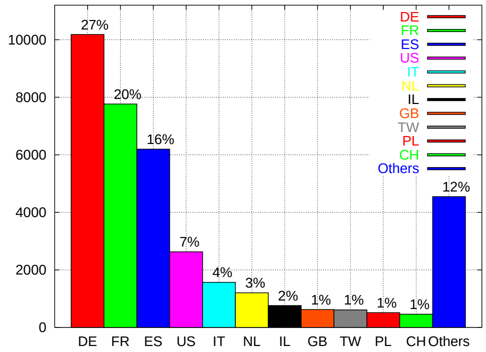

Figure 1: Distribution of peers per country

3 Workload Properties

Our crawler provides us with the content lists of each peer that replied with a non empty list of files. The first set of results presents the properties of the workload in terms of popularity, distribution between various types of files and sharing profiles of peers.

As opposed to previous studies which gathered requests from a particular location – such as a university [6]–, we gather a map of contents over a large number of countries (see Figure 1).

3.1 Replication Patterns

While previous studies evaluated the popularity as the number of requests per file, we measure a file's popularity by its replication degree. Results match former observations on file popularity; *i.e.*, a few files are extremely replicated while a large number is replicated once and approximately 100,000 files are present in two caches or more. Figures 2 and 3 respectively present the replica distribution of files for all files and for various types of files, per file rank. We observe similar properties between the eDonkey workloads and the Kazaa workloads presented in [6], and in particular after an initial flat region, the popularity distributions have a clear linear trend on log-log plots.

3.2 Sharing and Distribution

Figures 4 and 5 display respectively the CDF of number of files and the disk-space that each

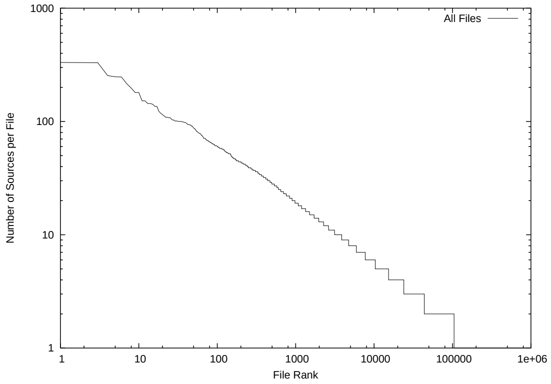

Figure 2: Replica distribution of files (900,000 files)

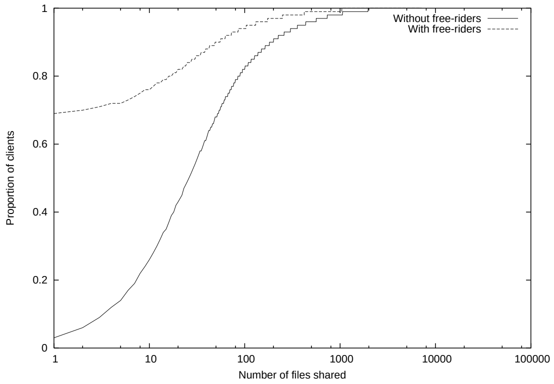

Figure 4: The number of files *offered* per client

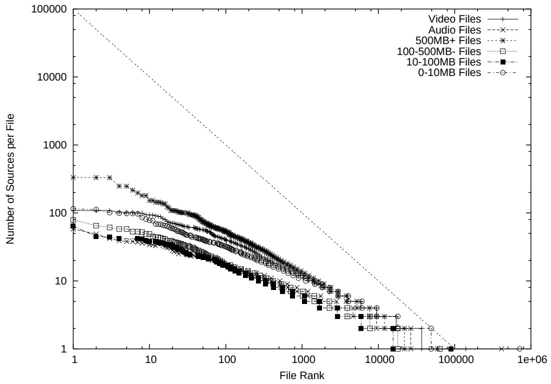

Figure 3: Replica distribution for different classes of files

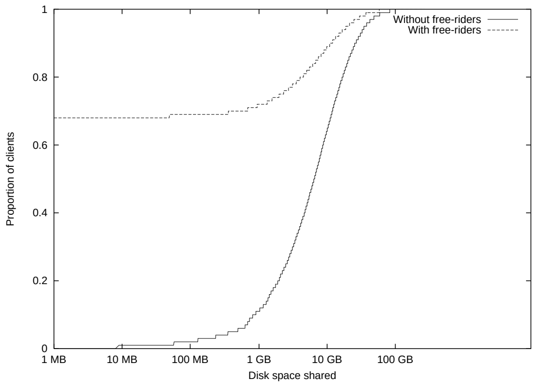

Figure 5: The disk-space shared per client

peer offers to the network. While the proportion of free-riding is still significant (68 %), most of the remaining clients share few files (between 10 and 100), but large files (between 1 and 50 GB).

We have also investigated the proportion of files of a given type share in the sytem, measured both in number of files and disk-space. We observe that multimedia files, audio and video, dominate: Audio files represent the largest number of files (48% against 16% for video), while in terms of size video files are dominant (67% against 16% for audio). Other measures show that *MP3* files dominate in number, while *AVI* files largely dominate in terms of size. Although such distribution is expected and matches results obtained on different workloads, it is interesting since it leads to the exploitation of two different types of clustering: geographical and interest-based clustering.

3.3 Clustering

For clarity, we consider only the two major types of files, audio and video files, leading to different opportunities in terms of clustering exploitation.

Geographical Clustering and Video Files. It seems reasonable to assume that the traffic generated by video files dominates largely in the network. For such files, the latency and traffic induced by the search process are negligible as compared to the download phase. Figure 6 shows that, for more than 65 % of the files, half of the sources are located in one country, and for half of them, 75 % of the sources are in one country. This shows the presence of geographical clustering, *i.e.*, peers requesting a given video file may in a large proportion of cases download it from peers in their own country, thus achieving low latency and network usage compared to downloading it from a randomly chosen peer.

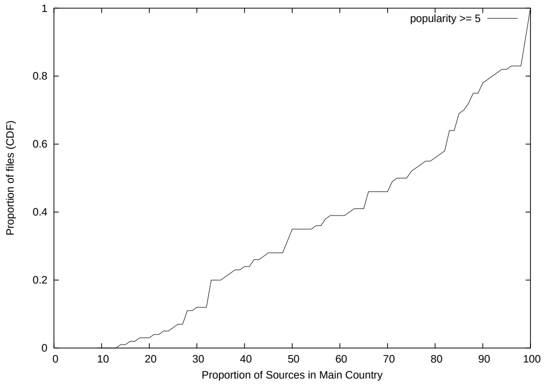

Figure 6: Proportion of sources in one country (for video files with more than 5 sources).

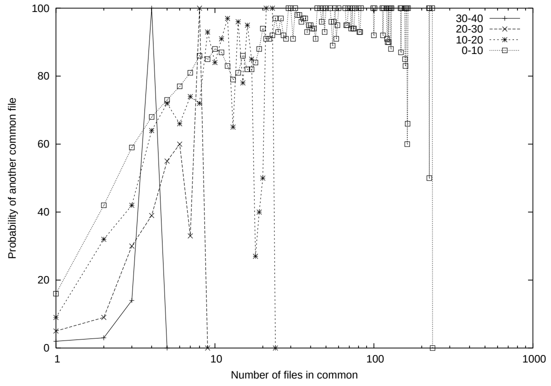

Figure 8: Audio file clustering (according to popularity)

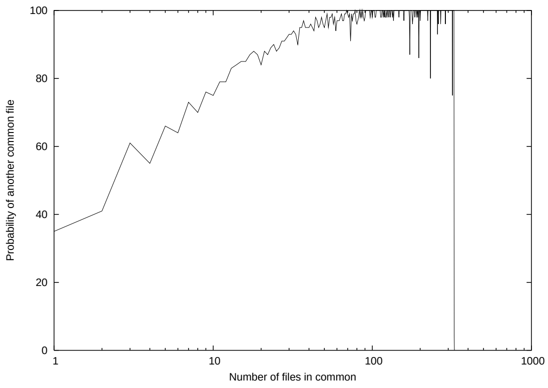

Figure 7: Clustering (all files)

Interest-based Clustering and Audio Files. As 48 % of the shared files are audio files, it is reasonable to assume that most requests are for such files. Since audio files are small, the latency and traffic generated by a request is not negligible as compared to the actual download. Performance of P2P search for such files can be greatly improved if we exploit interest-based locality [13, 4]: if two peers share interests, (in other words if the contents of their cache overlap significantly) the search mechanism can be significantly improved by having these peers connect to one another and first send their requests to each other.

Clustering Measurements. Figure 7 displays the clustering between every pair of peers and is measured as the probability that any two clients having at least a given number of files in common share (at least) another one. We observe that this curve is increasing very quickly with the number of files in common, up to a certain point (here, 325) that reflects the maximal cache overlap in the observed trace. As soon as some peers have a small number of files in common (say 10), the probability is high (approximatively 0.8) that they will have another one in common. This probability is very close to 1 for peers sharing over 50 files in common.

Figure 8 presents the same clustering information, calculated for sub-classes of files, namely for audio files, and depending on their popularity. Results indicate that the clustering is pretty high for un-popular audio files (2-10 replicas) and in addition the maximal overlap between peers cache contents reach over 200 unpopular audio files. As the popularity increases, the overlap tends to decrease as well (30-40 replicas). However, the probability increases consistently and fairly quickly with the number of files in common.

We also evaluated the correlation between files, rather than between peers. We define the correlation between two files as the correlation between the fact that a randomly selected peer has either file. More precisely, this is the correlation coefficient between the indicator variables equal to 1 if a randomly chosen peer has the document, and zero otherwise. Figure 9 displays the cumulative distribution function of the maximal correlation coefficients of files to other files. By reading the value for a correlation coefficient

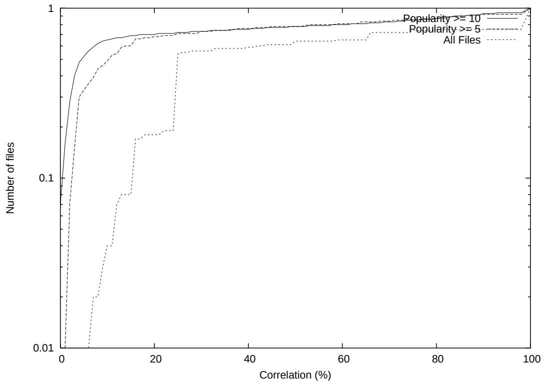

Figure 9: The CDF of maximal correlation to other files (for distinct file popularities)

value of 40 %, one thus finds that more than 60 % of the files have a correlation of more than 70 % with some other file, and hence are usually found in the same caches. This can be expected, when files are cdrom images of programs or movies in several parts, sets of cracks for programs, and un-popular MP3 tracks of popular albums.

Further interpretation of the results, as well as processing of the data is needed to get a better characterisation of interest-based clustering and how it might be exploited. Such extensions are currently under way.

4 Conclusion

Peer to peer file sharing is now the main component of Internet traffic. Yet, there have been only a few studies of workload measurements and even less consideration of clustering properties. In this paper, we presented the results of an evaluation we conducted on the most popular file sharing network in Europe, namely the eDonkey 2000 network, in an attempt to provide a map of the content shared by its peers. In particular, we focused on the geographical and the interest-based clustering properties for video and audio files respectively, of the observed workload. We demonstrated that there is a significant locality of interest for audio files, that could be used to improve the search mechanisms without relying on servers. The interest of this measurement study is three fold: First, to the best of our knowledge, it represents the first evaluation study focusing on clustering properties and providing such a map of contents. Second, we also aim at doing a more comprehensive study by looking at a broader spectrum of users from different countries, and not limited to a particular community. Finally, the workloads we evaluated can be used as input to evaluate peer-to-peer file sharing systems and contribute to the improvement of such systems.

References

[1] E. Adar and B. Huberman. Free riding on gnutella. *First Monday 5*, Oct. 2000.

[2] R. Bhagwan, S. Savage, and G. Voelker. Understanding availability. In *IPTPS'03*, Feb. 2003.

[3] Y. Chawathe, S. Ratnasamy, L. Breslau, N. Lanham, and S. Shenker. Making gnutella-like p2p systems scalable. In *SIGCOMM'03*, 2003.

[4] A. Crespo and H. Garcia-Molina. Semantic overlay networks for p2p systems. Technical report, Stanford University, 2003.

[5] F. L. Fessant. Mldonkey, a multi-network file-sharing client. http://www.mldonkey.net/, 2002.

[6] K. P. Gummadi, R. J. Dunn, S. Saroiu, S. D. Gribble, H. M. Levy, and J. Zahorjan. Measurement, modeling, and analysis of a peer-to-peer file-sharing workload. In *SOSP'03*.

[7] N. Leibowitza, A. Bergman, R. Ben-Shaul, and A. Shavit. Are file swapping networks cacheable? characterizing p2p traffic. In *WCW'02*.

[8] D. Plonka. Napster traffic measurement. Technical report, University of Wisconsin-Madison, 2000.

[9] A. Rowstron and P. Druschel. Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems. In *Middleware*, 2001.

[10] S. Saroiu, K. P. Gummadi, R. Dunn, S. D. Gribble, and H. M. Levy. An analysis of Internet content delivery systems. In *OSDI'02*, Dec. 2002.

[11] S. Saroiu and S. G. P. Krishna Gummadi. A measurement study of peer-to-peer file sharing systems. In *MMCN'02*, Jan. 2002.

[12] S. Sen and J. Wong. Analyzing peer-to-peer traffic across large networks. In *SIGCOMM'02 Workshop on Internet measurment*, 2002.

[13] K. Sripanidkulchai, B. Maggs, and H. Zhang. Efficient content location using interest-based locality in peer-to-peer systems. In *INFOCOM'03*.

[14] I. Stoica, R. Morris, D. Karger, F. Kaashoek, and H. Balakrishnan. Chord: A scalable peer-to-peer lookup service for internet applications. In *SIGCOMM 2001*, San Diego, USA, Aug. 2001.

[15] G. Wearden. eDonkey pulls ahead in European P2P race. http://news.com.com/2100-1025_3-5091230.html, 2003.