Clustering in P2P exchanges and consequences on performances

Clustering in P2P exchanges

Jean-Loup Guillaume and Stevens Le Blond

LIAFA – CNRS – *Université Paris 7, 2 place Jussieu, 75005 Paris, France.*
Faculty of Sciences – Vrije Universiteit, De Boelelaan 1081A, 1081 HV Amsterdam, The Netherlands.

We propose here an analysis of a rich dataset which gives an exhaustive and dynamic view of the exchanges processed in a running eDonkey system. We focus on correlations in terms of data exchanged by sets of peers having provided or queried at least one common data. We introduce a method to capture these correlations (namely the data clustering), and study it in detail. We then use it to propose a very simple and efficient way to group data into clusters and show the impact of this underlying structure on search in typical P2P systems.

Keywords: peer-to-peer, clustering, graphs

1 Preliminaries

P2P networks such as KaZaA, eDonkey, Gnutella and more recently BitTorrent are nowadays the most bandwidth consuming applications on the Internet, ahead of Web traffic [SGD$^+$02]. Their analysis and optimisation therefore appears as a key issue for computer science research. However, the fully distributed nature of most of these protocols makes it difficult to obtain relevant information on their actual behavior. Since these behaviors have some crucial consequences on the performance of the underlying protocol (both in terms of answer speed and in terms of used bandwidth), it is a challenge of prime interest to collect and analyze such data. The observed properties may then be used to design efficient protocols.

Context

In the last few years, both active and passive measurements have been used to gather information on peers behaviors in P2P networks. These studies gave evidence for a variety of properties which appear as fundamental characteristics of such systems. Among them, let us notice the high ratio of free-riders, the heterogeneous distribution of the number of queries by peer, and recently the presence of semantic clustering in file sharing networks (see [FHKM04, SMZ02, GLB05] and references therein). This last property captures the fact that if two peers are interested in some data, they will probably share interest in other data. Connecting directly such peers, makes it possible to take benefit from this semantic clustering to improve search algorithms and the scalability of the system.

In [FHKM04], the authors propose a protocol based on this idea, which reaches very high performances. It however relies on a static classification which can hardly be maintained up to date. Another similar approach consists in the addition of a link in a P2P overlay between peers exchanging files [SMZ02, VKMvS04]. This has the advantage of being very simple and allows a significant improvement of the search process. In [FHKM04, HKFM04] the authors use traces of a running eDonkey network, obtained by crawling caches of a large number of peers. They study some statistical properties like replication patterns, various distributions, and clustering based on file types and geography, but they also use these data to simulate protocols and to evaluate their performances. The use of actual P2P traces where previous works used models (whose validity is hard to evaluate) is an important step. However, the large number of free-riders, as well as other measurements problems, makes it difficult to evaluate the relevance of such traces. Moreover, these measurements miss the dynamic aspects of the exchanges and the fact that fragment of files are made available by peers *during* the download of the files.

Framework and contribution

Our work lies in this context: we collected traces using a modified eDonkey server, which made it possible to grab accurate information on *all* the exchanges processed by a large number of peers during a significant portion of time. The server handled up to 50 000 users simultaneously and we collected 24 hour traces. The size of a typical trace at various times is given in Figure 1. See [BLG04] and references therein, for details on the measurement procedure, on the protocol and on the properties of our traces.

A natural way to encode the gathered data is to define a bipartite graph $\mathcal{Q} = (P,D,E)$, called *query graph*, where P is the set of peers in the network, D is the set of data and $E \subseteq P \times D$ is the set of undirected edges, where $\{p,d\} \in E$ if and only if the peer p is *active* for the data d (p is interested in or is a provider of d). Notice that this graph evolves during time.

In order to analyze our data, we will also consider the (weighted) *data graph* $\mathcal{D} = (D,E,w)$ obtained from the query graph $\mathcal{Q}$ where D is the set of data, $E \subseteq D \times D$ is the set of undirected edges, where $\{d_1,d_2\} \in E$ if and only if there exists a peer active for both d_1 and d_2 in $\mathcal{Q}$. Finally, w is a weight function over the nodes and the edges such that $w(d)$ is the total number of data exchanged by peers active for d, and $w(d_1,d_2)$ is the number of data exchanged by peers active for both d_1 and d_2.

	6h	12h	18h	24h
peers	26187	29667	43106	47245
data	187731	244721	323226	383163
links in $\mathcal{Q}$	811042	1081915	1571859	1804330
links in $\mathcal{D}$	12238038	20364268	31522713	38399705

Figure 1: Time-evolution of the basic statistics for $\mathcal{Q}$ and $\mathcal{D}$

In this paper, we first focus on the *data clustering*, which captures how much the exchanges processed by two sets of peers are similar. We then show that these properties have significant impact on the efficiency of searches in the network and therefore may be used in the design of efficient P2P protocols.

2 Data clustering analysis

Our aim is now to analyze similarities between data in terms of exchanges processed by peers active for them. In particular, given two data u and v exchanged by a given peer p we are interested in the number of other common data exchanged by peers active for u or v. This can be measured using the following parameter:

$$c(u,v) = \frac{w(u,v)}{w(u)+w(v)-w(u,v)}$$

Indeed, the two data u and v induce an edge $\{u,v\}$ in D, the weight $w(u,v)$ is nothing but the number of common data exchanged by peers active for u or v, and the expression $w(u)+w(v)-w(u,v)$ gives the total number of data exchanged by peers active for u or v. Finally, $c(u,v)$ therefore measures how much these exchanges overlap. Notice that its value is between 0 and 1.

The value of $c(u,v)$ may however be strongly biased: if one of the two nodes has a high weight and the other a low one, then the value would be very low. For example, if a data with an high popularity[†] is connected to an unpopular one, then the clustering will probably be low, even if the few data exchanged by the lowest population are completely included in the large set of data exchanged by the highest one.

In order to capture such dissymmetric behaviors, we will also consider the following statistical parameter:

$$\overline{c}(u,v) = \frac{w(u,v)}{\min(w(u),w(v))}$$

which still lies in $[0;1]$ but is always larger than $c(u,v)$ and does not have this drawback. For instance, in the case described above the obtained value is 1. We will call $c(u,v)$ the *clustering* of $\{u,v\}$ and $\overline{c}(u,v)$ its *min-clustering*.

[†] The popularity of a data is the number of peers active for that data.

Figure 2 shows the time-evolution of the distributions of $c(u,v)$ and $\overline{c}(u,v)$, respectively. First notice that the general shape of these distributions is very stable along time, which indicates that the observations we will derive are not biased by the timescale or date considered.

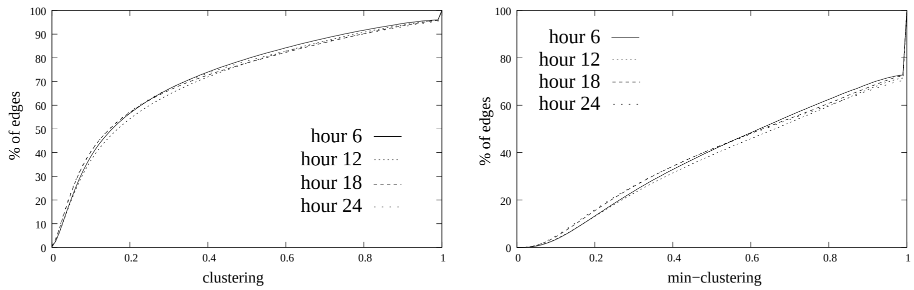

Figure 2: Time evolution of the $c(u,v)$ cumulative distribution (left) and of the $\overline{c}(u,v)$ cumulative distribution (right)

Now let us observe (Figure 2 left) that around 60% of the edges always have a clustering lower than 0.2. This may indicate that the overlap of exchanges is not as high as expected. However, this may be a consequence of the fact that both the peer activity and the data popularity are very heterogeneous: there are very active peers while most of them are not, and there are very popular data while most are not. This induces in $\mathcal{D}$ many links between data of very different popularity and a low clustering.

This can be corrected using the distribution of min-clustering (Figure 2 right): only 15% of the edges have a min-clustering lower than 0.2 while for nearly 60% higher or equal to 0.5, which indicates that the overlap is indeed high. For instance, 30% of the overlaps between all exchanges are a complete inclusion.

Such results could denote the presence of a hierarchy among these exchanges. While few popular data form the roots of $\mathcal{D}$, a large number of less popular data have their exchanges highly included into the upward level. Notice that if confirmed, such a property would give a solution to build dynamically a multicast tree from a P2P overlay.

3 Consequences on searching

Following several previous works [FHKM04, SMZ02, VKMvS04, HKFM04]), one may wonder if the properties highlighted in previous section may be used to improve search in P2P systems. Suppose that each peer knows the peers active for the same data as itself. Then, when p sends a query for d, it first queries these peers. If one of them provides d, then it sends it directly to p. In this case, the clustering effect has been used and the data was found using only one hop search.

The time-evolution of this hit ratio is plotted in Figure 3 (left). Despite it is quite low in the first few minutes (due to the server bootstrap), the ratio quickly converges to a value close to 45%.

To have a better understanding, the percentage of hits after 24h is correlated with the corresponding percentage of peers, the volume of requests they generated and the replication pattern of the queried data at the Figure 3 (right). The first thing to notice is that nearly 25% of the peers do not find *any* data using the proposed approach. This is quite surprising given the fact that we observed that 50% of *all* the requests are routed successfully with one hop. This can be understood by observing that this 'null hit' population generated only 7% of the requests and then only slighly influenced the high hit rate previously observed. Additionally, the queried data appear to be very slightly replicated at the time they were asked. This low volume of requests together with the low replication pattern constitutes an explanation to the null hit rate: these peers are not active for enough data nor enough replicated ones to find new ones using the one hop protocol.

On the other hand, more than 10% of the peers have a *perfect* success rate. One could think that such a result would imply a prohibitive amount of requests, Figure 3 (right) indicates that it's not the case but the percentage of requests is rather correlated with the number of peers who proceed them. Notice however that data found this way appear to be very highly replicated (the population being active for these data at the time they were asked represents 15% of the peers active for other queried data) which explains the high success rate. Finally, notice that the average peer's success rate increases to nearly 60% if the 'null hit' population is removed from the calculus.

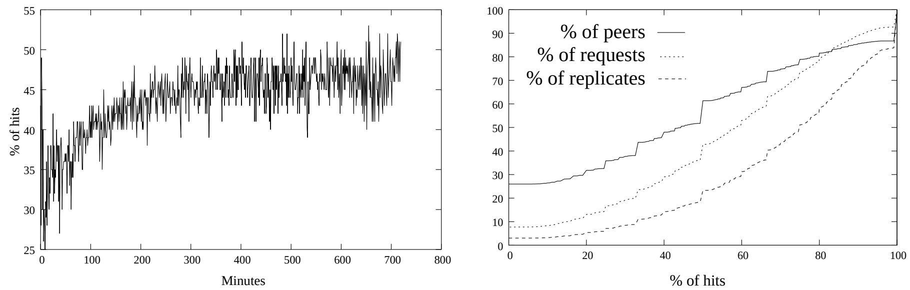

Figure 3: Time evolution of the hit % using the one hop protocol (left) and cumulative distribution of the % of peers, the associated % of requests they generated, the % of replication of the queried data and the % of hits they obtained after 24h using the one hop protocol (right)

Maintaining one group knowledge for each data exchanged in the system might be too costly. However, such a knowledge is already used for other issues. In BitTorrent, for instance, some centralized components, known as trackers, periodically distribute sources of peers asking for a given data. One exchange overlay by data is then maintained to guarantee fairness together with optimal transfer rates [Coh03]. Such a mechanism could also be used to propagate the requests of the one hop protocol we just described to provide efficient routing facilities in BitTorrent without more efforts.

4 Conclusion

In this contribution, we proposed simple statistical parameters to capture the correlations between the set of peers active for a given data. We used these parameters to confirm that semantic clustering can be used to improve search algorithms. These parameters can also be used to define a very simple and efficient way to compute data clusters, which partially succeed in capturing similarities between data. Notice also that we focused here on *data*, but the same kind of approach might be fruitful with peers.

Finally, let us insist on the fact that the analysis of large real-world traces like the one we presented here is only at its beginning, and that much remains to understand from it. The lack of relevant statistical parameters (concerning for example the dynamics of the trace), and of efficient algorithms to deal with such traces are among the main bottleneck to this, but studies like the one we presented here show that simple methods can already bring much information.

References

[BLG04] Stevens Le Blond, Matthieu Latapy, and Jean-Loup Guillaume. Statistical analysis of a p2p query graph based on degrees and their time evolution. IWDC'04, 2004.

[Coh03] Bram Cohen. Incentives build robustness in bittorrent, 2003.

[FHKM04] F. Le Fessant, S. Handurukande, A.-M. Kermarrec, and L. Massoulié. Clustering in peer-to-peer file sharing workloads. 3rd International Workshop on Peer-to-Peer Systems (IPTPS), September 2004.

[GLB05] Jean-Loup Guillaume, Matthieu Latapy, and Stevens Le Blond. Clustering in p2p exchanges and consequences on performances. 4th International Workshop on Peer-To-Peer Systems (IPTPS'04), 2005.

[HKFM04] S. Handurukande, A.-M. Kermarrec, F. Le Fessant, and L. Massoulié. Exploiting semantic clustering in the edonkey p2p network. 11th ACM SIGOPS European Workshop (SIGOPS), September 2004.

[SGD+02] S. Saroiu, K. Gummadi, R. Dunn, S. Gribble, and H. Levy. An analysis of internet content delivery systems, 2002.

[SMZ02] K. Sripanidkulchai, B. Maggs, and H. Zhang. Efficient content location using interest-based locality in peer-to-peer systems. Internet Measurement Workshop 2002, November 2002.

[VKMvS04] S. Voulgaris, A.-M. Kermarrec, L. Massoulie, and M. van Steen. Exploiting semantic proximity in peer-to-peer content searching. FTDCS, 2004.