Toward a Global File Popularity Estimation in Unstructured P2P Networks

Manel Seddiki*, Mahfoud Benchaiba‡
*,‡ University of Sciences and Technology Houari Boumediene
Computer Science Department, LSI laboratory
Algiers, Algeria
* e-mail: sed.manel@gmail.com
‡ e-mail: benchaiba@lsi-usthb.dz

***Abstract*—In unstructured P2P networks, replicating most popular files is one of mechanisms, which improve file lookup performances, such as lookup delay and success rate. However, measuring global file popularity is a challenging task because this estimation must consider requests of all peers for this file whereas in unstructured P2P networks like Gnutella, the peer has no global view of the network. Some researches have been done to measure this parameter. Nevertheless, this estimation is still away from reality because the peer, which calculates file popularity, doesn't consider file popularity estimations of the other peers. In this paper, we try to define a way to calculate a global file popularity based on local estimation of the peer and estimations done by the other peers participating in the network. Our first simulation results reinforce our theoretical formulas and show that our measurement is closer to the real one. More details will be provided and simulation tests will be added in our future contributions.**

***Keywords*—Unstructured P2P networks, global file popularity, file lookup,request packets, replication.**

I. Introduction

Peer-to-peer (or P2P) networks came to replace client/server systems and were developed over Internet in recent years. The basic idea of P2P is to link users in order to exchange information without using any intermediate server. Thus, P2P network is a distributed system of interconnected peers, which are both clients and servers. The P2P paradigm was firstly used for file-sharing applications such as Napster [1] and Gnutella [3], which allow users to lookup, share and download files.

Napster uses a server which indexes all the information about peers and their files. If a peer wants to lookup for a file, it sends a request to the server, which connects it directly with peers storing this file. The server facilitates the lookup procedure and improves the lookup latency, but it is the weakness of the system because if it breaks down, the whole system stops. Gnutella came after Napster and erased centralization idea. Indeed, Gnutella works on an unstructured P2P network architecture, where there is no server and each peer must know the other peers participating in the P2P network and their shared content by itself. A peer wishing to lookup for a shared content, such as a file, broadcasts its request to all its neighbors, which do the same with their neighbors until the file is found or the Time To Life (TTL) expires. This technique is denoted as flooding [3]. However, the flooding main drawback is the high overhead that causes a scalability issue. Many alternatives to flooding have been proposed to make file lookup technique more efficient, such as using probability based on previous lookup results ([4] and [5]), using progressive TTL called Expending Ring such as [6] or using Random walk technique such as [7].

Another way to improve file lookup performances in P2P unstructured networks is replication, as presented in [8], [9], [10], and [11], which consists in the replication of most popular files in other peers to ensure their availability, increase lookup success rate and decrease lookup hops and delay. Performances of these replication strategies depend on the popularity parameter precision. Indeed, the closer is the popularity estimation from reality, the better is the replication strategy performance. As a consequence and for our point of view, the file popularity measurement in such replication strategies is then crucial to decide which files have to be replicated. However, most of these strategies don't focus on this measurement and briefly define file popularity calculation based only on local estimation of the peer. This is maybe due to the fact that in P2P unstructured architectures, the peer is blind and has no global view of the network and this makes global popularity estimation a challenging task. In this paper, we focus completely on this issue and try to define the file popularity notion and four evident criteria that the file popularity estimation must respect. After that, we propose a way to calculate the file popularity according to and respecting those creteria. This calculation is based both on local estimation of the peer and estimations done by the other peers participating in the network. Indeed, considering the estimations of the other peers allows having a global-like estimation of the popularity which is closer to the reality than the local estimation.

This paper is organized as the following: In Section II, we introduce some interesting researches which calculate file popularity used in variety of contexts, such as content replication strategies and file lookup enhancement. In Section III, we describe our approach in detaills. We begin first by describing the P2P network architecture and environment that we consider in our approach then, we describe our file cache structures and define the popularity notion according to our point of view. After that, we explain our file popularity measurement and finally, we discuss some points. In Section IV, we introduce simulation environnement, describe the different simulation tests and compare our estimated popularity with the real popularity. In the end of this paper, we give a brief summary of this paper's content and next contributions to finalize our

ISBN: 978-1-61208-305-6

work.

II. Related work

Despite of the file popularity importance in the replication and file lookup area, there are no consistent investigations in calculating a global file popularity, which is close to the real popularity. However, many replication strategies, such as those in [8], [9], [10], and [11] proposed some simple popularity measurments. In [9], the file popularity measurement is simply obtained by counting the number of accesses of each file f as follows: Peer P2 asks for a file f from the peer P1 ; P1 is able to provide file f or its index; P2 accesses P1 to retrieve file f ; P1 increments f popularity as follows:

$$P_f = P_f + 1 \tag{1}$$

In [8], the Q-replication strategy defines a popular file as a file which is frequently accessed. Each peer maintains a table containing the file name and the file popularity. The file popularity for each file f is calculated as follows:

$$P_f(t+1) = P_f(t) + \eta \frac{R_f(t)}{N(t)} * 100 \tag{2}$$

R$_f(t)$ is the number of requests seen by the peer for the file f at time t , N(t) is the total number of requests received by the peer at time t and η is a constant variable. The popularity is updated according to (2) after a fixed total number of requests received.

Another way to estimate file popularity is described in [10]. In this paper, the popularity is defined as the request rate for a file f and it is calculated as follows:

$$P_f = \frac{R_f}{T} \tag{3}$$

R$_f$is the number of requests peer have seen for the file f and T is the amount of time the peer has been up. The popularity is updated each time the peer receives a request for file f.

In [11], a dynamic data replication strategy is propsed. Indeed, to improve grid system performances, authors propose a dynamic strategy to replicate data in several sites of the grid considering crash failures in the system. The strategy is based on 2 parameters: Availibility and popularity of data. The popularity of the data f is calculated in this paper as follows:

$$P_f = \frac{R_f}{N} \tag{4}$$

R$_f(t)$ is the number of requests demanding f and N is the total number of all requests.

In our opinion, file popularity estimation has to respect four criterions:

- The popularity value is a rate and must be between 0 and 1.
- As the popularity depends on external actors (in our case, file requests), it must increase when request rate for this file is high and decrease when it is low. Let us take for example an artist-painter: His popularity depends on its fans (external actors) , it increases when its fans request highly its paintings and it decreases when not.
- Popularity value must be influenced explicitly or implicitly by time and this criterion is related to the previous point.
- Popularity must be based on global knowledge of requests circulating in the network.

All of [8], [9], [10], and [11] are based only on local estimations of the peer in popularity measurement. They don't acquire a global knowledge about the file popularity. In [8] and [9] and according to (1) and (2), the popularity measurement is cumulative, which means that the value will never decrease. Moreover, it is not between 0 and 1. In [8] and according to (1), popularity is not influenced by time and in [10], the popularity is defined as the number of requests for the file f by time unit. This leads to simply request rate and not popularity estimation. We conclude that criteria mentionned above are not all respected by [8], [9], [10], and [11]. In this paper, we try to consider all those criteria to provide a file popularity definition and calculate its estimation in order to make it close to the real value.

III. Our contribution

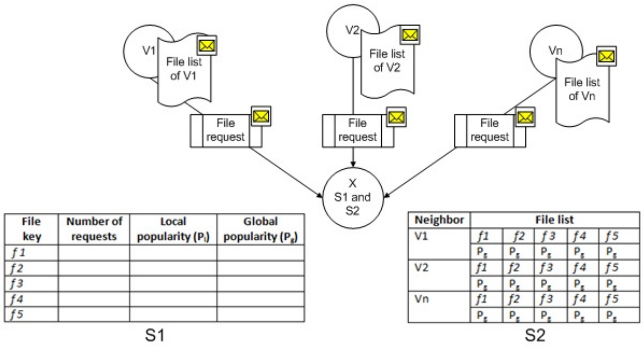
In this section, we describe our file popularity measurment which is based on both local popularity measurment of the node and popularity measurement of its neighbors. The idea is to have a global-like knowledge about the file by using neighbors which did the same with their neighbors and so on.

A. P2P network environement

We consider unstructured P2P architecture where peers index their own files and have no knowledge about the other shared files in the network and their locations at the beginning. Our file popularity measurement operates during file lookup phase and each peer is supposed to have at minimum, one neighbor.

B. Files cache

Each peer X participating in the P2P network maintains 2 structures denoted by S1 and S2 as shown in Figure.1. The first structure S1 stores local files and files discovered from file request packets passed through X. Each entry of S1 contains information that the peer knows about the file which, are the file key, number of requests passed through X for this file, local popularity and global popularity calculated by X. All explanation about how to calculate local and global popularity will be given in next section. The second structure S2 stores all the files's popularities of X's neighbors. S1 and S2 will be used to extract all necessary information needed in the computation of file popularity. S1 is initialized by adding local files of peer X with number of requests=0, local popularity=0 and global popularity=0. New entries in S1 are added when the peer X discovers new information about a file in the request packet passed through it and increment number of request by 1 for the concerned file. Moreover, peer X exchanges periodically its file list with its neighbors. S2 is initialized and updated when X receives this list.



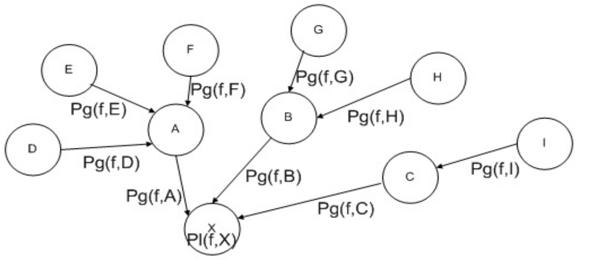
File key	Number of requests	Local popularity (P_l)	Global popularity (P_g)
f 1			
f 2			
f 3			
f 4			
f 5			

S1

Neighbor	File list				
V1	*f1*	*f2*	*f 3*	*f 4*	*f 5*
	P_g	P_g	P_g	P_g	P_g
V2	*f1*	*f2*	*f 3*	*f 4*	*f 5*
	P_g	P_g	P_g	P_g	P_g
Vn	*f1*	*f2*	*f 3*	*f 4*	*f 5*
	P_g	P_g	P_g	P_g	P_g

S2

Figure. 1: File cache structure



Pg(f,X)=*function*(Pl(f,X),Pg(f,A),Pg(f,B),Pg(f,C)) where:
Pg(f,A)=*function*(Pl(f,A),Pg(f,D),Pg(f,E),Pg(f,F))
Pg(f,B)=*function*(Pl(f,B),Pg(f,G),Pg(f,H))
Pg(f,C)=*function*(Pl(f,C),Pg(f,I))

Pl is local popularity estimation
Pg is global popularity estimation

Figure. 2: Global popularity estimation scheme for peer X

C. *File popularity definition*

For the best of our knowledge, a file is popular if it is highly requested in the network. Several definitions of file popularity have been disscussed in the related work section. We define the file popularity as the ratio between the number of requests for the file *f* and total number of requests in the entire network formulated as follows:

$$P(f, X) = \frac{number\,of\,requests\,for\,the\,file\,f}{total\,number\,of\,all\,requests} \qquad (5)$$

The real file popularity estimation presented in (4) can be only calculated by a global observer, which has a global view of the entire P2P network. However, peers have no global view in unstructured P2P network. Indeed, each peer is blind and has only local knowledge about the file. This local information is not enough to have a real estimation of file popularity. Thus, our goal is to find a way to bring global-like information about files and include it with local information to have file popularity estimation closer to the real estimation. In our approach, peers benefit from the knowledge of the other participating peers through neighbors. In fact, the peer calculates file popularity based on its own knowledge and knowledge of its neighbors. Knowledge of neighbors is obtained based on the own knowledge of neighbors and the one of their neighbors, and so on, as it shown in Figure. 2 . In this way, all peers cooperate to provide a global view of the file in the network and thus, estimate a file popularity closer to the real one.

D. *File popularity estimation*

In this section, we define our file popularity measurement. It is composed of two major steps. The first step is the local popularity estimation, which is based on the local knowledge of the peer about the file. Local knowledge is obtained by exploiting file request packets passed through the peer. The second step is global popularity estimation, which is based on the local popularity estimated in the first step and the global popularity estimated by direct neighbors. At the beginning, the local and global file popularities are defined by 0 for each file *f*. Local file popularity is updated when the peer receives a request packet form its neighbors and global file popularity is updated when the peer receives a file list from its neighbors.

1) Local file popularity estimation: It consists on calculating file popularity based on local knowledge of the peer. The local popularity of the file *f* for the peer X denoted by $\mathrm{P}_l(f, X)$ is defined as the ratio of known requests for the file *f* denoted by R_f to all known requests denoted by R. It is formulated as follows:

$$P_l(f, X) = \frac{R_f}{R} \qquad (6)$$

R and R_f are obtained from structure S1. The local popularity is updated each time a peer receives a request packet.

2) Global file popularity estimation: Local popularity estimation is not enough to reflect the real value. Indeed, we need to have a global estimation of *f*'s popularity by considering both the local estimation formulated in (5) and global estimations of neighbors. It is calculated as follows:

$$P_g(f, X) = \frac{(P_l(f, X) + \sum_{j=1}^{|V|} P_g(f, V_j)}{(|V| + 1)} \qquad (7)$$

Where $|V|$ is number of neighbors of peer X which, have calculated global popularity of file *f* , $\mathrm{P}_l(f, X)$ is local popularity of the file *f* for the peer X and $\mathrm{P}_g(f, V_j)$ is global popularity of the file *f* for the neighbor V_j such as $1 \leq j \leq |V|$. $|V|$ and $\mathrm{P}_g(f, V_j)$ are obtained from S2.

E. *Discussion*

The global popularity is calculated when the peer receives file list from its neighbors. This calculation is done in two ways:

- The first way (which, we consider in this paper) is the periodic list reception from neighbors. In this case, the challenge is to select the suitable delay because if it is too small, the overhead increases in the network due to high exchange of file lists and if this delay it is too large, this may result in imprecision on popularity estimation and lack of updates concerning file requests and peers disconnections.
- The second way is the on-demand list reception, which means that if the peer wants to calculate file popularity for replication or for other purpose, it requests its neighbors for the file list. The advantage on asking for file list on-demand is that neighbors send only the

TABLE I: Simulation parameter

Simulation time	1000s
Average neighbors	3
Number of nodes	100
File list delay	40s
Node join and departure	Lifetime churn with lifetime=600s
request load	1 request for random file per 60s

concerned file and not all files and this will decrease file list size but the drawback is the time wasted on waiting for the file information to be received.

Our popularity estimation is based both on local knowledge of the peer and global-like knowledge acquired through neighbors as explained in previous sections. This estimation is bounded by 0 and 1. Moreover, it increases when request number for the concerned file is high comparing with the other requests and decreases when not and time influences the estimated value implicitly through those requests. Hence, the four criterions are respected.

IV. Simulation

In order to compare our file popularity estimation with the real popularity value, we implemented our file popularity algorithm and a global observer algorithm on OverSim [12] with Omnet++ [2]. The P2P network is composed of 100 peers, which may join and leave according to lifeTimeChurn=600s as shown in table I. Each peer in the network has a random number of local files limited to 100 maximum and enriches its structures S1 and S2 through file list exchanged between neighbors and request packets passed through the peer. A peer sends a request for a file choosen rondomly every 60s. We chose one file with key=E88 from the network to observe its popularity evolution. Our initial results are obtained by comparing our popularity estimations with the global observer estimations. In Figure.3, the thick line with square symbols represents real file popularity evolution calculated by the global observer and the other thin lines represent file popularity estimated by some peers participating on the network according to our approach. Thus, Figure.3 shows that all peers estimate popularity values that match closely with the real popularity calculated by the global observer. This is due to the cooperation between all peers in order to allow having to each single peer, a global-like view of the file. A best view of this match is represented in Figure.4 where the general bahaviour of the system represented with triangle symbols match closely with the global observer behaviour represented with square symbols. These simulation results reinforce our theoretical formulas and prove that our file popularity estimation is efficient in an unstructured P2P network.

V. Conclusion

Estimating real file popularity in unstructured P2P networks is a hard task because peers are blind and have no global view of the network resources. In our point of view, calculating file popularity value, which is close to reality must respect four criterions : It must be bounded by 0 and 1; it must increase and decrease according to external actors; it must be influenced by time implicitly or explicitly; it must be based on a global-like knowledge about the concerned file. Several researches proposed to measure the file popularity, but not all the four criteria were considered.

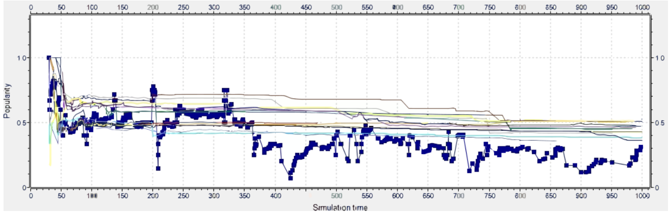


Figure. 3: Real popularity vs estimated popularity of file E88

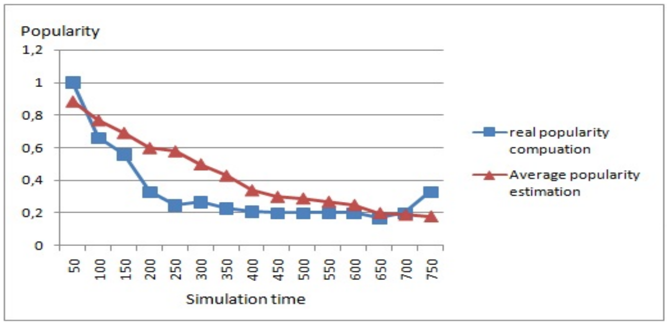


Figure. 4: Real popularity vs estimated popularity of file E88

In this paper, we define file popularity and we propose a measurement for it respecting the four criteria. Our first simulation results reinforce our theoretical formulas and show that our measurement matches closely with the real one. These initial results prompt us to investigate more about this rate. More details will be provided and simulation tests will be added, such as the impact of the search rate on the popularity deviation in our future contributions.

References

[1] (1999) Napster. [retrieved:march , 2013]. [Online]. Available: http://www.napster.co.uk.

[2] (2002) Napster. [retrieved:april , 2013]. [Online]. Available: http://www.omnetpp.org.

[3] M. Ripeanu and I. Foster, "Mapping the gnutella network, Internet Computing," IEEE, vol. 6, 2002, pp. 5057.

[4] R. Gaeta and M.Sereno, "Generalized probabilistic flooding in unstructured peer-to-peer networks, Parallel and Distributed Systems," IEEE Transactions, vol. 22, December. 2011, pp. 2055 2062.

[5] S. Margariti, "A novel probabilistic flooding strategy for unstructured peer-to-peer networks, 15th Panhellenic Conference, " September. 2011, pp. 149153.

[6] Q. Lv, P. Cao, E. Cohen, K. Li and S. Shenker, "Search and replication in unstructured peer-to-peer networks, Proceedings of the International Conference on Supercomputing," June 2002, pp.22-26.

[7] C. Gkantsidis and A. Saberi, "Random walks in peer-to-peer networks, Proc. of IEEE INFO-COM," vol. 1, March. 2004, pp. 130140.

[8] SM.Thampi and K. Sekaran, "Review of replication schemes for unstructured P2P networks,"' arXiv preprint arXiv:0903.1734, no. March. 2009, pp. 67.

[9] S. Mohammadi, H. Pedram, and A. Farrokhian, "An enhanced data replication method in p2p systems, Journal Of Computing," vol. 2, November 2010, pp. 7882 .

[10] J. Kangasharju, W. Ross, and D. Turner, "Optimal content replication in p2p communities, Manuscript" 2002.

[11] B. Meroufel and G. Belalem, ”Dynamic replication based on availability and popularity in the presence of failures, Journal of Information Processing Systems,” Vol.8. June. 2012, pp. 263278.

[12] I. Baumgart and S. Krause, ”Oversim: A flexible overlay network simulation framework, 2007 IEEE Global Internet Symposium,” May. 2007, pp. 7984.