

# Définition et diffusion de signatures sémantiques dans les systèmes pair-à-pair

Raja Chiky\*, Bruno Defude\*\*, Georges Hébrail\*

\* GET-ENST Paris

Laboratoire LTCI - UMR 5141 CNRS  
Département Informatique et Réseaux  
46 rue Barrault, 75634 Paris Cedex 13  
Email: chiky@enst.fr, hebrail@enst.fr

\*\*GET-INT

Département Informatique  
9 rue Charles Fourier, 91011 Évry cedex  
Email: bruno.defude@int-evry.fr

**Résumé.** Les systèmes pair-à-pair (peer-to-peer, P2P, égal-à-égal) se sont popularisés ces dernières années avec les systèmes de partage de fichiers sur Internet. De nombreuses recherches concernant l'optimisation de la localisation des données ont émergé et constituent un axe de recherche très actif. La prise en compte de la sémantique du contenu des pairs dans le routage des requêtes permet d'améliorer considérablement la localisation des données. Nous nous concentrons sur l'approche PlanetP, faisant usage de la notion de filtre de Bloom, qui consiste à propager une signature sémantique des pairs (filtres de Bloom) à travers le réseau. Nous présentons cette approche et en proposons une amélioration : la création de filtres de Bloom dynamiques, dans le sens où leur taille dépend de la charge des pairs (nombre de documents partagés).

## 1 Introduction

Pour la recherche, le partage et l'échange de ressources (données, programmes, services), le modèle pair-à-pair constitue une alternative au modèle client/serveur. Les pairs peuvent à la fois offrir (rôle serveur) et demander (rôle client) des ressources. Il existe de nombreuses architectures des systèmes pair-à-pair, se basant sur des techniques différentes de localisation des données, qui se traduisent par des méthodes différentes de routage des requêtes. Pour améliorer la localisation d'une ressource recherchée par un pair, on ajoute de l'information aux tables de routage des requêtes : il peut s'agir du contenu des pairs, de l'historique de leurs requêtes, ou des concepts qu'ils traitent...

La difficulté rencontrée lors de l'intégration de la sémantique du contenu des pairs, est de déterminer un espace de représentation commun à tous les pairs du réseau. Quelques systèmes tels que SON (*Semantic Overlay Network*)(Crespo et al., 2002) utilisent des concepts définis a priori pour résoudre ce problème. Mais cette solution ne s'applique qu'à un domaine précis.

Pour pallier cet inconvénient, PlanetP (Cuenca-Acuna et al., 2002) utilise une signature sémantique pour représenter le contenu de chaque pair. Cette signature est définie par une structure de données appelée filtre de Bloom (Bloom, 1970).

Notre travail s'inscrit dans le cadre du projet RARE mené au sein du GET (Groupement des Ecoles des Télécommunications). Dans ce projet, plusieurs approches sont étudiées comme l'utilisation des filtres de Bloom, la propagation efficace des index via des algorithmes de Gossiping ou encore l'apprentissage sur les requêtes passées par des mémoires associatives. Dans ce papier, nous nous intéressons à l'utilisation des filtres de Bloom dans PlanetP et proposons une amélioration permettant de réduire la taille des filtres de Bloom, et par conséquent de faciliter leur diffusion à travers le réseau pair-à-pair, tout en maintenant les performances de la recherche d'information. L'approche est validée par des expérimentations sur les collections de données suivantes : CACM, CISI, MED et CRAN de SMART (Buckley, 1985).

Les sections 2 et 3 de cet article décrivent l'approche PlanetP et le fonctionnement du filtre de Bloom. La section 4 définit notre approche et la section 5 présente les différentes expérimentations menées.

## 2 Filtres de Bloom

### 2.1 Définition

Un filtre de Bloom (Bloom, 1970) est un tableau de bits qui permet de tester d'une manière rapide l'appartenance d'un terme à un certain ensemble de termes (ceux d'un document ou d'un ensemble de documents). Le filtre de Bloom consiste en deux composants : un ensemble de  $k$  fonctions de hachage et un vecteur de bits d'une taille  $m$  donnée. Toutes les fonctions de hachage sont configurées de telle sorte que leurs intervalles correspondent à la taille du vecteur. Par exemple, si le vecteur de bits est de taille 200, alors toutes les fonctions de hachage doivent retourner une adresse entre 1 et 200. Les fonctions de hachage garantissent que les adresses générées sont réparties de façon équiprobable sur toutes les valeurs possibles.

Pour introduire un terme dans un filtre de Bloom, on calcule les valeurs des fonctions de hachage et on active (on met à 1) les bits du vecteur correspondants.

Pour tester l'appartenance d'un terme  $t$  à un ensemble  $Y$  de termes introduits dans le filtre de Bloom, on lui applique les fonctions de hachage. Si au moins un des bits est à 0 alors le terme  $t$  n'appartient pas à  $Y$ . Par contre, si tous les bits sont à 1 alors  $t$  appartient probablement à  $Y$  avec un taux de faux positif moyen donné par la relation suivante :

$$f = (1 - e^{-kn/m})^k \quad (1)$$

Où  $m$  est la taille du vecteur du filtre de Bloom,  $n$  le nombre de termes indexés et  $k$  le nombre de fonctions de hachage.

Pour minimiser ce taux, on choisit un nombre de fonctions de hachage  $k$  respectant la relation suivante :

$$k = \ln 2(m/n) \quad (2)$$

## 3 PlanetP

### 3.1 Définition

PlanetP est un système pair-à-pair permettant de faire de la recherche textuelle sur le contenu des documents stockés par les pairs. Le contenu de chaque pair est représenté de manière compacte à l'aide d'un filtre de Bloom décrivant les termes des documents stockés par celui-ci. Les filtres de Bloom sont distribués dans le réseau en utilisant un algorithme de propagation : chaque pair a donc une vision résumée des contenus disponibles dans d'autres pairs. Afin de supporter la recherche par le contenu, PlanetP utilise deux structures de données présentes sur chaque pair :

- Un index local : Chaque pair indexe localement de façon exacte les termes contenus dans ses propres documents.
- Un index global : Chaque pair dispose d'une liste d'autres pairs, associés chacun à leur filtre de Bloom, permettant de donner au pair une vision partielle et approchée du contenu global du réseau. Les filtres de Bloom sont échangés entre les pairs en utilisant un algorithme de propagation appelé « Algorithme de Gossiping ». Ce qui permet à chaque pair d'améliorer sa connaissance du réseau.

### 3.2 Recherche dans PlanetP

La recherche d'information est basée sur le modèle vectoriel. PlanetP propose une approximation à la mesure TFxIDF, qui nécessiterait une connaissance de tous les mots du réseau. Cette mesure adaptée aux P2P est l'Inverse Peer Frequency IPF, pouvant être calculée à l'aide des informations locales à chaque pair. La mesure IPF pour un terme  $t$  est donnée par :

$$IPF_t = \log(1 + N/Nt) \quad (3)$$

Où  $N$  est le nombre de pairs connus dans le réseau par le pair effectuant la recherche et  $N_t$  le nombre de pairs parmi ceux-ci ayant les documents contenant  $t$ .

Un nœud qui reçoit une requête cherche dans son index local. S'il ne peut pas honorer la requête, il calcule les rangs des pairs de son index global. Pour donner un rang aux pairs, on utilise l'expression suivante :

$$R_i(Q) = \sum_{t \in Q \text{ et } t \in BF_i} IPF_t \quad (4)$$

Où  $Q$  est la requête,  $BF_i$  le filtre de Bloom du pair  $i$  et  $t$  un terme de la requête. La requête est alors propagée aux pairs de plus grand rang.

## 4 Filtre de Bloom dynamique

Il existe des pairs qui contiennent plus de documents que d'autres. Généralement, la répartition des documents dans les pairs suit une loi de Zipf (Goh et al., 2005). Le nombre de termes à indexer dans les filtres de Bloom varie donc selon le nombre de documents de chaque pair. Une idée est de créer des filtres de Bloom de taille dynamique, dans le sens où la taille

du filtre d'un pair dépend de son nombre de documents. Nous fixons une taille maximale pour les filtres de Bloom et un nombre maximum de fonctions de hachage. Chaque fonction sera affectée à un intervalle d'une partition du filtre de Bloom. Par exemple, nous fixons :

- une taille maximale de 10 000 bits ;
- 4 fonctions de hachage (chaque fonction donnera une adresse sur 2 500 positions)
  - 1ère fonction : 1 → 2 500 ;
  - 2ème fonction : 2 001 → 5 000 ;
  - 3ème fonction : 5 001 → 7 500 ;
  - 4ème fonction : 7 501 → 10 000.

Soit  $d_i$  le nombre de documents du pair  $i$ , on choisit :

- si  $d_i \leq 40$  : une seule fonction de hachage et un filtre de Bloom de 2 500 bits ;
- si  $41 \leq d_i \leq 60$  : deux fonctions de hachage et un filtre de Bloom de 5 000 bits ;
- si  $61 \leq d_i \leq 80$  : trois fonctions de hachage et un filtre de Bloom de 7 500 bits ;
- si  $d_i \geq 81$  : quatre fonctions de hachage et un filtre de Bloom de 10 000 bits ;

Cette méthode nous permet de réduire d'environ 50% la taille des filtres de Bloom, ce qui facilitera leur diffusion à travers le réseau. Le fait de réduire la taille des filtres de Bloom n'affecte pas le taux de faux positif car le nombre de fonctions de hachage a été choisi selon l'équation ( 2).

## 5 Expérimentations et résultats

Pour nos expériences, nous avons utilisé les quatre collections de documents utilisées par PlanetP pour son évaluation (requêtes et jugements de pertinence associés). La table 1 présente le contenu de ces collections : elles sont composées de fragments de textes et résumés, et sont relativement petites en taille. Nous les avons préalablement traitées grâce à l'outil de recherche d'information SMART (Buckley, 1985), afin d'extraire les mots lemmatisés, leurs fréquences et d'éliminer les mots vides.

Pour tester les performances de notre approche, nous avons utilisé les métriques standard, rappel(R) et précision (P), définies comme suit pour une requête Q :

$R(Q)$  = nombre de documents pertinents retournés / nombre de documents pertinents dans la collection

$P(Q)$  = nombre de documents pertinents retournés / nombre de documents retournés

	CACM	CISI	CRAN	MED
Nb. documents	3204	1460	1400	1033
Nb. termes uniques	3029	5755	2882	4315
Nb. moyen de termes par document	18.4	38.2	49.8	46.6
Nb. requêtes	64	112	225	30
Nb. moyen de termes par requête	9.3	23.3	8.5	9.5
Nb. moyen de documents pertinents par requête	15.3	27.8	8.2	23.2

**TAB. 1** – Collections de documents utilisées dans les expérimentations et leurs caractéristiques

La similarité entre une requête  $Q$  et un document  $D$  est calculée comme suit :

$$sim(Q, D) = \frac{\sum_{t \in Q} IPF_t \times \log(1 + f_{D,t})}{\sqrt{|D|}} \quad (5)$$

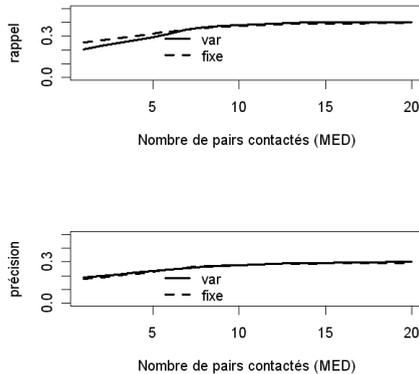
Où  $f_{D,t}$  est le nombre d'apparitions du terme  $t$  dans  $D$  et  $|D|$  le nombre de termes dans  $D$ .

Pour mesurer le rappel et la précision dans les collections, nous distribuons aléatoirement les documents sur 20 paires virtuels selon la loi de Zipf sans redondance (un document n'est présent que sur un seul pair). Nous construisons la signature de chaque pair en utilisant chacune des méthodes suivantes :

- Filtre de Bloom fixe construit avec deux fonctions de hachage (comme dans le système PlanetP) avec une taille de 10 000 bits ;
- Filtre de Bloom variable. Nous utilisons les configurations de l'exemple de la section 4.

La taille des filtres de Bloom a été choisie afin d'assurer un taux de faux positif inférieur à 5%. Le rang des paires est calculé par l'équation (4) pour chacune des requêtes des collections puis ceux-ci sont triés par ordre décroissant de la mesure du rang. Ensuite  $p$  paires jugés pertinents sont sélectionnés,  $p$  variant de 1 à 20 paires. Les documents contenus dans ces  $p$  paires sont triés par la mesure de similarité selon l'équation (5). Nous extrayons les 30 documents de similarités les plus élevées et mesurons le taux de rappel et de précision. Par manque de place, nous ne présentons dans la figure 1 que les courbes obtenues pour la collection MED, des résultats similaires étant obtenus sur les autres collections.

La figure 1 montre le taux de rappel et de précision en fonction du nombre de paires contactés.



**FIG. 1** – (a)Rappel et (b)Précision pour la collection MED : var correspond au cas du filtre de Bloom variable et fixe au cas du filtre de Bloom avec deux fonctions de hachage.

Nous observons à travers ces courbes que l'utilisation d'un filtre de Bloom dynamique n'altère pas les taux de rappel et de précision par rapport à un filtre de Bloom fixe.

## 6 Conclusion

Le système PlanetP permet de faire de la recherche textuelle de documents dans un environnement distribué, en proposant un même espace de représentation partagé par tous les pairs, sous la forme de filtres de Bloom de taille fixe. Nous avons proposé de rendre cette taille dynamique, en fonction du nombre de documents stockés par chaque pair. Les expérimentations montrent que la dynamique ne détériore pas les performances obtenues dans le cas classique (filtre de Bloom fixe). La bande passante utilisée pour la propagation des filtres de Bloom peut ainsi être réduite, ou bien le taux de leur diffusion à travers le réseau peut être augmenté, afin d'enrichir les index globaux. L'évaluation chiffrée de ces gains reste à réaliser, par simulation des échanges entre pairs, en utilisant le simulateur également développé dans le projet RARE.

## Références

- Bloom H. (1970). Space/Time Trade-offs in Hash Coding with Allowable Errors. Commun. ACM 13, 7 (Jul. 1970), 422-426.
- Buckley C. (1985). Implementation of the SMART Information Retrieval System. Technical Report. UMI Order Number : TR85-686. Cornell University.
- Crespo A. et H. Garcia-Molina (2002). Semantic Overlay Networks for P2P Systems. Technical report, Computer Science Department, Stanford University.
- Cuenca-Acuna F., C. Peery, P. Martin et D. Thu Nguyen (2002). PlanetP : Using Gossiping to Build Content Addressable Peer-to-Peer Information Sharing Communities. Department of Computer Science, Rutgers University. 12<sup>th</sup> IEEE International Symposium on High Performance Distributed Computing (HPDC-12).
- Goh S., P. Kalnis, S. Bakiras et K. Tan (2005). Real Datasets for File-Sharing Peer-to-Peer Systems. DASFAA, 201-213.

## Summary

Peer-to-peer systems (P2P) have become popular these last years with sharing files on Internet. Much research has emerged concerning the optimization of data localization, and constitutes a very active research area. Taking into account semantic aspects improves considerably data localization. We based our study on the PlanetP approach, which uses the notion of Bloom Filter consisting on propagating peers' semantic signatures (Bloom Filters) through the network. We present this approach and extend it with an improvement : the creation of dynamic Bloom Filters, their size depends on the load of peers (number of shared documents).