Evaluation de Requêtes Flexibles dans un Contexte Non-Centralisé : Une Approche Basée sur les Résumés Distribués

Abdelkader Alem[1], Allel Hadjali[2]

[1] Département d'Informatique, Université Ibn Khaldoun BP 78, 14000 Tiaret, Algérie
alemainhadid@yahoo.fr
[2] ENSSAT/IRISA, Université Rennes 1, 6 rue de kérampont - BP 80518,
22305 Lannion Cedex, France
hadjali@enssat.fr

Résumé. L'étude présentée dans cet article traite le problème d'évaluation des requêtes flexibles dans un contexte distribué, illustré par un système P2P. L'idée suggérée est de n'envoyer la requête qu'aux sources de données qui sont susceptibles de fournir les meilleures réponses à la requête, ce qui nécessite d'évaluer la pertinence de chaque source au moyen de son résumé. L'approche proposée consiste en deux étapes. La première étape concerne la construction d'un index de routage global (et distribué), qui décrit les données des différentes sources, afin de trouver l'ensemble des pairs pertinents pour la requête posée. La seconde étape a trait à la recherche des réponses satisfaisant au mieux cette requête.

Mots-clés: Requête flexible, système P2P, index de routage, résumé de donnée.

1 Introduction

Au cours de cette dernière décennie, le paradigme des systèmes distribués, en particulier les systèmes dits P2P, est devenu très populaire en permettant le partage des ressources et l'échange d'information entre des millions d'utilisateurs. De plus, les P2P offrent de nombreux autres avantages comme l'auto-organisation, l'autonomie et le "passage à l'échelle". Parmi les systèmes P2P les plus connus, on peut citer Gnutella [10], Napster [11]. Cependant, un des problèmes majeurs dont souffrent ces systèmes est la localisation des sources (ou pairs) pertinentes (c.-à-d., celles contenant les réponses qui satisfont au mieux les besoins de l'utilisateur). Ce problème a fait l'objet de plusieurs études et de nombreuses techniques efficaces ont été proposées pour la localisation de données pertinentes dans les systèmes P2P [6].

Les index de routage [6] font partie des techniques proposées dans la littérature pour une évaluation efficace (en évitant la stratégie d'inondation) des requêtes dans les réseaux P2P. Rappelons qu'un index de routage est une structure de données (avec un ensemble d'algorithmes) qui, étant donnée une requête sur un pair, retourne la liste de pairs voisins ordonnés selon leur pertinence à la requête considérée. Ces index

permettent donc de ne renvoyer la requête qu'aux pairs qui sont les plus probables à fournir de réponses. On distingue deux catégories d'index : les index locaux et les index globaux. Dans Gnutella [10], chaque nœud maintient un index local sur les données qu'il possède (un nœud diffuse la requête à tous ses voisins dans le réseau). Un index global peut être centralisé ou distribué, par exemple, Napster [11] stocke un index global de toutes les données sur un pair central (ou super pair), alors que Chord [16] distribue un index global sur tous les nœuds du réseau.

Par ailleurs, les requêtes à préférences est un thème de recherche qui a aussi suscité un intérêt croissant ces dernières années, voir par exemple [2][5][7]. Les requêtes dites Skyline [2] ont reçu une attention particulière de la part d'un grand nombre de chercheurs dans la communauté des bases de données. Cependant, la plupart des travaux proposés ont été réalisés dans un environnement centralisé. Relativement, peu d'études existent dans un cadre décentralisé [9][12][13][18][19]. Mentionnons, par exemple, les travaux de Hose et al. [9] et de Zinn [19] qui ont proposé des approches pour le traitement des requêtes skyline dans des systèmes P2P. Dans ces travaux, les auteurs utilisent des index de routage appelés DDS (Distributed Data Summaries). Ces index sont basés sur une structure d'arbre particulière, appelée QArbre (traduction de l'anglais de QTree). Cette structure est une combinaison d'histogramme et d'une autre structure d'arbre (dite R-Tree). Dans chaque pair p, on maintient un résumé de toutes les données qui peuvent être accessibles par envoi d'une requête à tous les voisins de p.

Dans cet article, le problème considéré concerne l'évaluation des requêtes flexibles [3] dans un système distribué. Ces requêtes permettent aussi d'exprimer des préférences au moyen de prédicats flous (comme "jeune", "cher", etc.) dont la satisfaction est graduelle. Dans ce cadre, le résultat d’une requête n’est plus un ensemble "plat" d’éléments mais un ensemble où chaque élément est associé avec un degré de satisfaction.

Pour autant que nous le sachions, il n'existe pas de travaux sur cette problématique dans la littérature, excepté ceux proposés dans [4][8] où l'aspect évaluation des requêtes n'a pas été suffisamment traité. L'étude que nous suggérons combine l'approche décrite dans [19] pour la construction d'index de routage et le modèle de résumé introduit dans [15] pour la définition du contenu de l'index. En particulier, une stratégie d'évaluation de requêtes flexibles dans un environnement P2P est proposée. Elle permet de ne renvoyer que les réponses qui satisfont au mieux la requête posée.

Le reste de l'article est organisé comme suit. La section 2 introduit un exemple de référence servant à illustrer notre approche. La section 3 présente l'approche basée sur les QArbres pour la construction d'index de routage. En section 4, on décrit le modèle de résumé SaintEtiq. La section 5 est consacrée à l'approche proposée pour la construction d'index global et distribué. La section 6 discute l'évaluation de requêtes flexibles dans un contexte P2P. Un exemple illustratif est décrit dans la section 7. La section 8 conclut l'article et esquisse quelques directions pour de futurs travaux.

2 Exemple de Référence

L'exemple suivant est utilisé pour illustrer notre approche. Il s'agit de 3 magasins de vente (de cameras) répartis sur trois sites formant un réseau P2P (avec les sites 1 et 3 comme nœuds feuilles et le site 2 comme nœud central). Chaque site maintient sa propre base de données. Chaque tuple (caméra) est décrit par deux attributs *(prix, qualité)*, voir Tableau 1.

Tableau 1. Ensemble de caméras réparties sur trois sites.

Site 1			**Site 2**			**Site 3**		
Modèle	Prix	Qualité	Modèle	Prix	Qualité	Modèle	Prix	Qualité
S1	165	7,2	C1	90	5	X1	295	9,3
S2	275	8,1	C2	260	6,5	X2	310	12,8
S3	270	8,2	C3	375	7,9	X3	540	14,1
S4	369	13,6	C4	55	4,6	X4	330	8,1
S5	635	10,1	C5	410	14	X5	720	16,1
S6	350	10,1	C6	560	15,3	X6	100	7,9
S7	290	9,5	C7	521	15,1	X7	160	7,1
S8	395	8,3	C8	730	16,5	X8	360	10,3
S9	412	14	C9	820	17	X9	290	8
S10	537	14,7	C10	610	16,5	X10	270	7,8
S11	300	12,3				X11	340	8,9
S12	149	5,5						

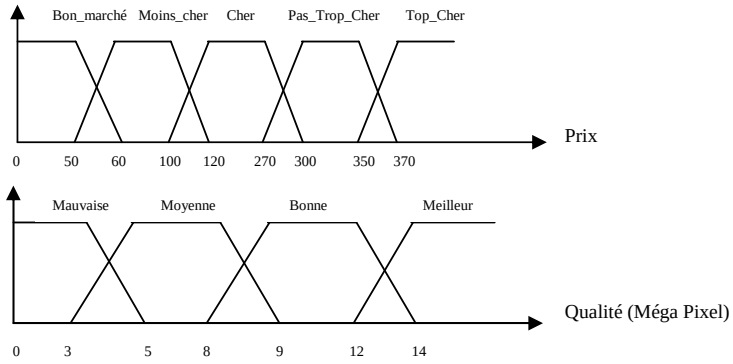

Fig. 1. Variables linguistiques décrivant les attributs "*Prix*" et "*Qualité*"

Dans le Tableau 1 la colonne "Modèle" est utilisée juste pour l'identification des tuples (le prix est exprimé en euros et la qualité concerne la résolution d'une caméra, elle est mesurée en termes de nombre de pixels). On suppose aussi disponible des partitions floues sur les attributs *"Prix"* et *"Qualité"*, voir Figure 1. Par exemple, les

prédicats flous "*Pas_Trop_Cher*" et "*Bonne*" sont représentés par les f.a.t[1] (300, 350, 30, 20) et (9, 12, 1, 2) respectivement. Voir [3] pour plus de détails.

3 Index de Routage : L'Approche de Zinn

Le problème traité par Zinn dans [19] concerne l'évaluation des requêtes Skyline dans un environnement P2P. La solution proposée comprend deux phases essentielles : (i) Construction d'un index de routage basé sur une structure hiérarchique dite QArbre; (ii) Evaluation des requêtes en exploitant l'index de routage construit. Dans ce qui suite, on décrit brièvement la première étape en présentant tout d'abord la structure de QArbre.

3.1 Structure de QArbre

Une structure de QArbre [19] est un arbre où chaque nœud correspond à une boîte multidimensionnelle rectangulaire (appelée MBB pour Minimum Bounding Box). Un MBB est le plus petit rectangle contenant de l'information sur les données se trouvant dans le sous-arbre (resp. nœud) si le nœud n'est pas une feuille mais un nœud interne (resp. si le nœud est une feuille). Cette information est de nature statistique comme, par exemple, le nombre de données accessibles via le nœud. Les feuilles sont appelées des "paniers" (buckets). Plus la taille d'un panier est réduite, plus il est plus facile de décider si ses données sont pertinentes pour répondre à une requête posée ou non.

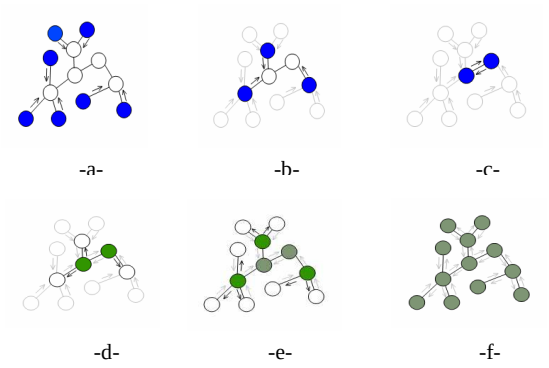

Fig. 2. Processus de construction de l'index

3.2 Principe de construction de l'index

Le QArbre est utilisé comme structure de base pour la compression des données du réseau. Le réseau est constitué d'un ensemble de pairs où chaque pair est relié à un petit ensemble de voisins et peut envoyer des messages à tous ses voisins. Un

[1] f.a.t signifie fonction d'appartenance trapézoïdale. Elle est représentée par un quadruplet (A, B, a, b) où [A, B] (resp. [A-a, B+b]) exprime le noyau (resp. le support).

algorithme distribué pour la construction de l'index est proposé dans [19]. La Figure 2 illustre les différentes étapes de cette construction dans un réseau P2P (avec 13 pairs). La Figure (Fig. 2-a) montre la phase d'initialisation : tous les nœuds feuilles (colorés en bleu) envoient de l'information sur leurs données locales (résumées au moyen de "buckets"). Puis les pairs ayant reçu des messages de tous leurs voisins sauf un (appelé N) résument leurs données locales avec toutes les données de leurs voisins (sauf évidemment N) dans des structures QArbres pour les envoyer ensuite au pair N (Fig. 2-b). Cette procédure est exécutée itérativement jusqu'à ce que les nœuds centraux reçoivent l'information de tous leurs voisins (Fig. 2-c). La dernière étape de l'algorithme consiste à faire l'opération inverse (l'information globale est diffusée dans le même chemin mais de haut en bas). Quand le dernier nœud feuille reçoit cette information de routage, l'algorithme se termine et ainsi tous les pairs possèdent des index de routage dits corrects (Fig. 2-d, e et f).

4 Résumé de Données : Le modèle SaintEtiQ

4.1 Construction du Résumé

SaintEtiq [15] est un modèle structuré pour les résumés de données. Il prend en entrée deux types d'informations : les données à résumer et les données relatives au domaine (appelées aussi connaissances sur le domaine). Ces connaissances sont constituées essentiellement de variables linguistiques définies sur les domaines d'attribut de la relation considérée (voir Figure 1).

Un résumé *z* peut être représenté par une paire *(Iz, Rz)* où *Iz* (resp. *Rz*) est appelé l'intension (resp. extension) de z. L'extension d'un résumé *z* est le sous ensemble des tuples impliqués dans *z*, alors que l'intension est la description linguistique de ces tuples. Par exemple, dans les Tableaux 2 et 3, l'intension du résumé Z_{f1} est I_{Zf1}= *{Cher, Moyenne}* et son extension R_{Zf1}= $\{Ct_1, Ct_{2,1}, Ct_{3,1}, Ct_{12,2}\}$.

Le modèle SaintEtiQ comprend deux grandes phases : i) la réécriture; et ii) l'organisation du résumé.

Tableau 2. Réécritures des Tuples du site 1 (Ch : Cher, Moy : Moyen, P_T_C : Pas très Cher)

Mod	Prix	Qualité	Tuples candidats
S1	16500	7,2	Ct1 (1.0/Ch, 1.0/Moy)
S2	27500	8,1	Ct2,1 (0.7/Ch, 1.0/Moy);
			Ct2,2 (0.3/P_T_C, 1.0/Moy)
S3	27000	8,2	Ct3,1 (1.0/Ch, 0.8/Moy);
			Ct3,2 (1.0/P_T_C, 0.2/Bon)
S4	36900	13,6	Ct4,1 (0.1/P_T_C, 1.0/Meil);
			Ct4,2 (0.9/T_C, 1.0/Meil)
…	…	…	…

4.1.1 Etape de réécriture

Cette étape permet au système de réécrire les tuples de la base de données avant que ceux-ci ne soient exploités par le service du résumé (un tuple réécrit est appelé tuple candidat, *Ct*). Elle donne naissance à un ou plusieurs tuples candidats qui peuvent être considérés comme des représentations linguistiques d'un tuple de la base.

Par exemple dans le Tableau 2, le tuple S4 est réécrit en deux tuples candidats Ct4,1 (0.1/P_T_Ch, 1.0/Meil) et Ct4,2 (0.9/T_Ch, 1.0/Meil). Le Tableau 2[2] donne quelques réécritures des tuples du site 1 des caméras.

4.1.2 Organisation du résumé

Cette étape consiste à organiser les résumés au sein d'une hiérarchie de manière à ce que le résumé le plus général soit placé au sommet de la hiérarchie et les résumés les plus spécifiques au niveau des feuilles. Le résumé racine décrit ainsi l'intégralité du jeu de données tandis que les feuilles ne résument qu'une partie plus limitée de la base. Voir Tableau 3 et Figure 3 (pour la hiérarchie de résumé associée au site 1).

Tableau 3. Classification des tuples candidats pour le site 1

Résumé Z	Intension	Tuples couverts par Z
Z_{f1}	(1.0/Ch, 1.0/Moy)	Ct_1, $Ct_{2,1}$, $Ct_{3,1}$, $Ct_{12,2}$
Z_{f2}	(1.0/P_T_C, 1.0/Moy)	$Ct_{2,2}$, $Ct_{11,1}$
Z_{f3}	(1.0/Ch, 1.0/Bon)	$Ct_{3,2}$, $Ct_{7,1}$
…	…	…
Z_{f7}	(1.0/P_T_C, 1.0/Bon)	Ct_6, $Ct_{7,2}$, $Ct_{11,2}$
…	…	…
Z_{13}	(1.0/P_T_C, 1.0/T_C, 1.0/Meil)	Z_{f4}, Z_{f5}
Z'_1	(1.0/Ch, 1.0/P_T_C, 1.0/T_C, 0,1/Mauv , 1.0/Moy, 1.0/Bon,)	Z_{11} , Z_{12} , Z_{f9}
Z_1	(1.0/Ch, 1.0/P_T_C, 1.0/T_C, 0,1/Mauv, 1.0/Moy, 1.0/Bon, 1.0/Meil)	Z_{13} , Z'_1

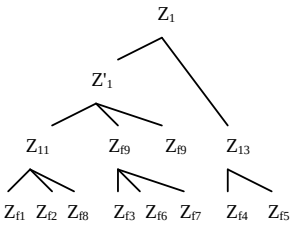

Fig. 3. Hiérarchie de résumé associée au site 1

4.2 Principe d'interrogation

L'interrogation d'une hiérarchie de résumé SaintEtiq a été abordée dans [17]. L'idée de base consiste à parcourir l'arbre des résumés afin de trouver les réponses qui satisfont la requête posée. Le résultat est donc un ensemble de résumés. Dans cette section, on rappelle les éléments essentiels de cette démarche.

Reprenons l'exemple des caméras (voir Section 2) et soit la requête flexible Q = *"Pas_Trop_Cher* $\wedge$ *Bonne"*. L'ensemble des étiquettes linguistiques apparaissant dans une requête sur l'attribut A_i sont dénotées par C_{Ai}. Les différents C_{Ai} sont regroupés dans l'ensemble C appelé *caractérisation initiale* liée à la requête posée. Par exemple,

[2] Faute de place, nous ne pouvons donner ici tout le contenu du Tableau 2 (resp. 3). Ces deux tableaux sont complètement décrits dans [1].

C = {Pas_Trop_Cher, Bonne} pour la requête *Q*. La sélection des résumés est réalisée par la fonction *Recherche* de l'algorithme 1. Cette fonction est récursive, elle prend en entrée un résumé *z* et la caractérisation initiale *C* de la requête. Elle retourne en sortie une liste de résumés L_{res}.

Dans l'algorithme 1 le test de correspondance, Corr(z, C), entre les caractères requis C_{Ai} des attributs de la requête et les descripteurs $L_{Ai}(z)$ extraits de l'intension de z, peut être de trois types :

Correspondance nulle : il existe au moins un attribut A_i pour lequel *z* ne montre aucun des caractères requis pour A_i.

Correspondance exacte : pour tous les attributs de la requête, le résumé *z* présente uniquement les caractéristiques recherchées, il est considéré comme résultat si *z* est une feuille.

Correspondance par excès : cette situation se présente lorsque le résumé *z* possède (sur un ou plusieurs attributs) plus de descripteurs que ceux de la requête, l'exploration du sous-arbre de racine *z* est donc nécessaire.

Algorithme 1. Fonction Recherche(z, C) [17]

```
Entrée : Résumé z et la caractérisation C
1. Lres := ϕ /* la liste est vide */
2. Si Corr(z, C) = excès Alors
3.  Pour chaque nœud fils zfils de z Faire
4.        Lres := Lres ∪ Recherche(zfils, C)
5.  Fin pour
6. Sinon
7.   Si Corr(z, C) = exacte Alors
8.        Si z est une feuille Alors
9.              ajouter (z, Lres)
10.       Sinon
11.             Lres := Lres ∪ Recherche (z, C)
12.       Fin si
13.  Fin si
14. Fin si
Résultat : Lres
```

5 Gestion des Résumés dans un Système P2P

Cette section est consacrée à la démarche proposée pour la construction d'un index global permettant le routage dans le réseau P2P. Il a été montré dans [17] qu'un résumé SaintEtiQ peut être considéré comme un index multidimensionnel. Dans cette étude, l'idée suggérée ici est de construire une hiérarchie de résumé (résumé global) qui décrit toutes les données du réseau P2P. L'approche proposée combine (i) l'algorithme décrit dans [19] pour la construction d'index de routage (cet algorithme présente l'avantage, d'une part, d'avoir une complexité temporelle acceptable et, d'autre part, de garantir une maintenance à moindre coût) ; et (ii) le modèle de résumé

de données SaintEtiq pour la définition du contenu de l'index (ce modèle conduit à un résumé exprimé en termes linguistiques et donc proche du langage de l'utilisateur).

Dans cette étude, le modèle P2P considéré est un système pair à pair décentralisé non structuré où la recherche se fait à l'aide des index de routage. Les hypothèses de base suivantes sont supposées vérifier :

- Les données sont arbitrairement distribuées sur les pairs.
- Chaque pair stocke et partage sa base de données locale.
- Tous les attributs impliqués dans la requête figurent dans les différentes bases de données distribuées.
- Chaque pair maintient un résumé local de sa propre base. Tous les pairs coopèrent pour construire un résumé global décrivant l'ensemble des données du réseau. On suppose aussi que les mêmes connaissances du domaine sont utilisées pour résumer les données de toutes les sources.
- Le problème d'hétérogénéité (et l'intégration de schémas) n'est pas considéré ici. Un schéma global est supposé.

5.1 Enrichissement des résumés

On considère ici qu'une hiérarchie de résumé sert comme un index multidimensionnel où un terme supplémentaire, P_z, est ajouté à la définition d'un résumé. Ce terme, dit Peer-extent [8], fournit l'ensemble des pairs qui ont des données décrites par le résumé *z*. On suppose aussi que dans la phase de réécriture, chaque tuple candidat contient un identificateur de son tuple original.

5.2 Procédure de construction de l'index

Cette procédure est largement inspirée de [19], la seule différence réside dans la manière de résumer les données. Dans notre cas, les données locales pour chaque pair sont résumées à l'aide du modèle SaintEtiQ. Ainsi, l'index global décrivant l'ensemble des données du réseau P2P est construit en suivant les étapes suivantes:

1. Une phase d'initialisation est faite sur chaque pair. L'ensemble des données locales est résumé à l'aide du modèle SaintEtiQ et un index local est créé pour chaque pair.
2. Tous les nœuds feuilles envoient une copie des informations sur leurs données (synthétisées sous forme d'une hiérarchie de résumés) vers ses voisins dans le réseau de P2P (Fig.2-a).
3. Tous les pairs ayant reçu des messages depuis leurs voisins sauf un (appelé N), fusionnent leurs informations locales (disponibles sous forme de résumé) avec toutes les informations reçues des différents voisins à l'exception du nœud N. Puis, ils envoient le résumé fusionné à N (Fig.2-b). L'opération de fusion suit exactement le modèle SaintEtiQ comme expliqué en Section 4.1 et elle est illustrée plus loin en Section 7.

4. L'étape 3 est répétée itérativement jusqu'à ce que le nœud central reçoive des informations de tous leurs voisins (Fig.2-c).
5. Le nœud central produit un résumé global (après la fusion de tous les résumés de ses voisins avec son résumé local). Ce résumé décrit toutes les données disponibles dans le réseau P2P, il sert comme un index sur les données du réseau (Fig.2-d).
6. La dernière étape consiste à faire l'opération inverse (le résumé global est diffusé dans le même chemin mais de haut en bas), et l'algorithme se termine quand les nœuds feuilles reçoivent l'index de routage global (Fig.3- e et f).

6 Stratégie d'Evaluation

Dans cette section, on présente une stratégie efficace pour localiser les pairs pertinents à une requête posée. Cette localisation est réalisée en exploitant l'index global (représenté par la hiérarchie des résumés). La solution proposée consiste à adapter l'algorithme 1 dans le sens où la réponse retournée n'est plus un ensemble de résumés, mais un ensemble de pairs. Ensuite, un algorithme de recherche des meilleurs tuples satisfaisant la requête est proposé. Cette recherche se fait en interrogeant la base de données originale par le biais d'un index local.

6.1 Localisation des pairs

Lorsqu'une requête est posée sur un pair, l'index (hiérarchie globale) est exploré à l'aide de l'algorithme 2 (une version adaptée de l'algorithme 1) donné ci-dessous.

Algorithme 2. Fonction SelectPair (z, C)

```
Entrée : Résumé z et la caractérisation C
 1. P_Q := ϕ /* la liste est vide */
 2. Si Corr(z, C) = excès Alors
 3.  Pour chaque nœud fils z_fils de z Faire
 4.          P_Q := P_Q ∪ SelectPair (z_fils, C)
 5.  Fin pour
 6. Sinon
 7.    Si Corr(z, C) = exacte Alors
 8.       Si z est une feuille Alors
 9.            ajouter (P_z, P_Q)
 10.      Sinon
 11.           P_Q := P_Q ∪ SelectPair (z, C)
 12.      Fin si
 13.   Fin si
 14. Fin si
Résultat : P_Q  /*la liste des pairs pertinents pour Q */
```

Rappelons que chaque résumé z contient dans sa description les pairs (P_z) qui sont décrits par ce résumé. Soit P_Q l'ensemble des pairs pertinents pour une requête Q. Le

principe de l'algorithme est : pour chaque nœud z du résumé qui correspond exactement à la requête, P_z est ajouté à l'ensemble P_Q. Puis, la requête est propagée vers l'ensemble des pairs contenus dans P_Q. Chaque pair qui a reçu le message, doit vérifier la satisfaction de la requête par rapport à ses données stockées localement (voir la sous-section suivante) et ainsi identifier les meilleurs tuples. Ensuite, il renvoie la réponse vers l'initiateur de requête. Lorsque le pair initiateur reçoit tous les résultats des pairs de l'ensemble P_Q, il filtre les résultats et renvoie seulement les réponses les plus satisfaisantes.

6.2 Evaluation de la requête sur chaque pair

Après que les pairs contribuant à la réponse d'une requête ont été identifiés. Il est nécessaire d'évaluer la requête flexible sur les données locales de chaque pair et donc la hiérarchie des résumés des données du pair (l'index local) est explorée pour trouver les résumés qui satisfont la requête. Comme les résumés feuilles regroupent les tuples candidats qui sont réécrits par les mêmes descripteurs linguistiques, on étend l'algorithme 1 pour qu'il retourne les enregistrements initiaux (tuples) ayant les degrés de satisfaction les plus élevés. L'algorithme 3 illustre cette procédure.

Algorithme 3. Fonction Recherche_Tuples (z, C)

Entrée : Résumé z et la caractérisation C
1. L_{Ct} := ϕ /* la liste des tuples candidats est vide */
2. L_{meil_Ct} :=ϕ /*liste des meilleurs tuples candidats vide */
3. $L_{réponse}$:= ϕ /* la liste des réponses est vide */
4. Si Corr(z, C) = excès Alors
5. Pour chaque nœud fils z_{fils} de z Faire
   L_{Ct} := L_{Ct} ∪ Recherche_Tuples (z_{fils}, C)
6. Fin pour
7. Sinon
8. Si Corr(z, C) = exacte Alors
9. Si z est une feuille Alors
10. Pour un tuple candidat Ct de z Faire
11. ajouter(Ct, L_{Ct})
    Fin pour
12. Sinon
13. L_{Ct} := L_{Ct} ∪ Recherche_Tuples (z, C)
14. Fin si
15. Fin si
16. Fin si
17. L_{meil_Ct} := BNL (L_{Ct})
18. Pour chaque Ct de L_{meil_Ct} faire
19. retourner le tuple original t
20. ajouter(t, $L_{réponse}$)
21. Fin pour

Résultat : $L_{réponse}$

La recherche est basée sur un test de correspondance comme dans l'algorithme 1, mais dans ce cas les résumés sont des index sur les données recherchées, quand un résumé feuille *z* correspond exactement à la requête, l'ensemble des tuples candidats couverts par *z* est ajouté à la liste des tuples candidats (L_{Ct}). Ensuite, la fonction *BNL*[3] retourne les meilleurs tuples L_{meil_Ct} parmi la liste L_{Ct}.

Une connexion à la base de données est nécessaire pour chaque tuple candidat *Ct* de la liste L_{meil_Ct}. Elle permet d'identifier les tuples de la base qui lui correspond. Le résultat final de l'algorithme est donc l'ensemble des tuples satisfaisant au mieux la requête.

7 Un Exemple Illustratif

Reprenons l'exemple de référence de la Section 2 et considérons la requête flexible *Q* concernant la recherche des caméras qui sont "*Pas_Trop_Cher*" et ayant une "*Bonne*" qualité. La solution proposée consiste premièrement à la construction de l'index de routage. Puis l'évaluation de la requête Q sur chaque source.

Pour construire l'index de routage, le pair 1 résume ses données par le résumé *Z1* comme expliqué en Section 4.1. De la même manière, les pairs 2 et 3 résument leurs données par *Z2* et *Z3*, voir Tableau 4 (pour plus de détails voir [1])

Tableau 4. Résumé global pour le réseau P2P.

Site 1 (Z_1)	(1.0/Ch, 1.0/P_T_C, 1.0/T_C, 0,1/Mauv, 1.0/Moy, 1.0/Bon, 1.0/Meil)
Site 2 (Z_2)	(0.5/B_Marché, 1.0/M_Ch, 0.5/Ch, 1.0/T_Ch, 1.0/Moyen, 1.0/Meil)
Site 3 (Z_3)	(1.0/M_Ch, 1.0/Ch, 1.0/P_T_C, 1.0/T_C, 1.0/Moy, 1.0/Bon, 1.0/Meil)
Résumé global (Z)	({0.5/B_Marché,1.0/M_Ch,1.0/Ch, 1.0/P_T_C, 1.0/T_C}, {0.1/Mauv, 1.0/Moy, 1.0/Bon, 1.0/Mei})

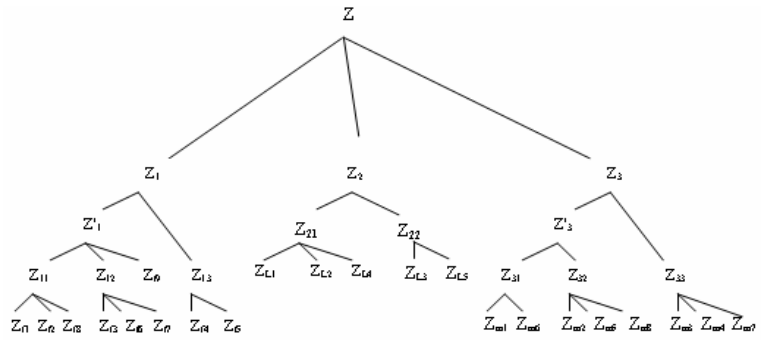

Fig 4 Hiérarchie de résumés pour le réseau P2P

Les deux pairs feuilles (site 1, site 3) envoient leurs résumés de données (index locaux) vers leur voisin (le nœud central site 2). Le pair 2 fusionne les résumés reçus

[3] basée sur le principe d'optimalité de Pareto, voir [2].

de ses voisins avec son propre résumé et délivre un résumé *Z* global (index global) qui décrit l'ensemble des données du réseau. Enfin, l'index *Z* (voir Fig 4) est renvoyé aux nœuds feuilles. Le Tableau 4 présente l'index local de chaque pair ainsi que l'index global construit.

Maintenant la requête Q est redirigée vers le pair 2. Celui-ci commence par exécuter l'algorithme 2, sur l'index global, pour déterminer les pairs pertinents. Premièrement, la racine *Z* correspond à la requête par excès, il faut tester ses nœuds fils, le nœud *Z2* à une correspondance nulle par rapport à la requête, donc le sous arbre de racine *Z2* est éliminé et ne peut être considéré par la suite, alors que les sous arbres de racines *Z1* et *Z3* (respectivement) sont considérés et ainsi de suite. Pour *Z1*, le résumé (nœud) Z_{f7} correspond exactement à la requête et son pair P_z= *{site1}*, alors que pour *Z3*, *Pz= {site3}*. Le résultat final de l'algorithme est P_Q = *{pair1, pair3}*.

La requête est donc routée vers les pairs 1 et 3. Chaque pair exécute l'algorithme 3. Le résultat de cet algorithme sur le pair 1 est le résumé Z_{f7} qui couvre les tuples candidats : {Ct_{61}, $Ct_{7,2}$, $Ct_{11,2}$} avec $Ct_{6,1}$ = (1.0/P_T_C, 0.9/Bon), $Ct_{7,2}$ = (0.8/P_T_C, 1.0/Bon) et $Ct_{11,2}$ = (1.0/P_T_C, 0.9/Bon). Les tuples de la base retournés sont donc {S6, S7, S11}. De la même façon, on peut vérifier que l'application de l'algorithme 3 sur le site 3 retourne les tuples candidats : {$Ct_{1,2}$, $Ct_{1,2}$} avec $Ct_{1,2}$ = (0.83/P_T_C, 1.0/Bon). Les tuples de la base retournés sont donc {X1, X11}.

Ensuite les deux pairs envoient leurs réponses au site initiateur de la requête, le site 2, qui fusionne ces résultats et délivre la réponse finale à la requête Q, notée $\sum_Q$, sous forme d'un ensemble flou de la forme :

$$\sum_Q = \{min(1.0,1.0)/S6,\ min(1.0,0.9)/S11,\ min(0.83,1.0)/X1,\ min(1.0,0.9)/X11\}$$
$$= \{1.0/S6,\ 0.9/S11,\ 0.9/X11,\ 0.83/X1\},$$

où chaque élément est associé avec un degré de satisfaction. On peut observer que la caméra *S6*, qui se trouve dans le site 1, satisfait totalement la requête *Q*. Par contre, la caméra *X11* du site 3 satisfait que partiellement la requête *Q* (avec un degré de 0.9).

8 Conclusion

Dans cet article, nous avons abordé le problème d'évaluation de requêtes flexibles dans un système P2P. Un index de routage global et distribué est proposé. Deux algorithmes ont également été étudiés. Le premier traite le problème de la localisation des pairs pertinents à une requête posée. Le second permet de retourner les tuples, contenus dans les sources associées aux pairs du système, qui satisfont au mieux la requête.

L'étude présentée ici est encore préliminaire et beaucoup de perspectives existent quant à des travaux futurs. Un premier travail à court terme concerne l'implémentation des algorithmes développés et la réalisation d'expérimentations afin de montrer la pertinence et l'efficacité de l'approche. Un second axe consiste à considérer d'autres modèles de résumé comme, par exemple, la typicité [4][14] des données contenues dans une source associée à un pair du réseau P2P.

Références

1. A. Alem, Contribution à l'étude de requêtes à préférences dans un système P2P, Mémoire de Magister, Dépt. d'Informatique, Université de Tiaret (Algérie), Novembre 2009.
2. S. Borzsonyi, D. Kossmann, K. Stocker, The skyline operator. Proc. 17th IEEE Inter. Conf. on Data Engineering, Heidelberg, 2001, pp. 421-430
3. P. Bosc, L. Liétard, O. Pivert, D. Rocacher, Gradualité et imprécision dans les bases de données, Paris, Editions Ellipses, 2004
4. P. Bosc, A. Hadjali, H. Jaudoin, O. Pivert, Flexible querying of multiple data sources through fuzzy summaries, Proc. of FlexDBIST'07, in conjunction with DEXA'07, Regensburg, Germany, September 3-7, pp. 350-354, 2007
5. J. Chomicki, Preference formulas in relational queries, ACM Transactions on Database Systems, 27, 2003, pp. 153-187.
6. A. Crespo, H. Garcia-Molina, Routing Indices for Peer-to-Peer Systems, Inter. Conference on Distributed Computing Systems, 2002.
7. A. Hadjali, S. Kaci, H. Prade, Database preferences queries - A possibilistic logic approach with symbolic priorities, Proc. of the 5th Inter. Symposium on Foundations of Information and Knowledge Systems (FoIKS'08), LNCS 4932, pp. 291–310, 2008.
8. R. Hayeky, G. Raschiay, P. Valduriezz, N. Mouaddib: Summary Management in P2P Systems, EDBT'08, 2008, Nantes, France
9. K. Hose, C. Lemke, K. Sattler, Processing relaxed skylines in PDMS using distributed data summaries, Proc. CIKM'06, USA, 2006
10. http://www.gnutella.com
11. http://www.napster.com
12. H. Li, Q. Tan, W. Lee, Efficient progressive processing of skyline queries in P2P systems, International Conference on Scalable Information Systems (INFOSCALE'06), 2006
13. E. Lo, KY Yip, K.I Lin, D.W. Cheng, Progressive skylining over Web-accessible databases, Data & Knowledge Engineering, 57, pp. 122-147, 2006
14. D. Merad Boudia, Contribution à l'étude de la typicité en vue de son application à l'interrogation flexible de bases de données, Dépt. d'Informatique, Université de Tlemcen (Algérie), Novembre 2009
15. G. Raschia, N. Mouaddib, A fuzzy set-based approach to database summarization, Fuzzy Sets and Systems 29(2), pp. 137– 162, 2002
16. I. Stoica, R. Morris, D. Karger, F. Kaashoek, H. Balakrishnan, Chord: A scalable Peer-To-Peer lookup service for internet applications. Proc. of the ACM SIGCOMM, pp. 149–160, 2001
17. W.A. Voglozin, G. Raschia, L.Ughetto, N. Mouaddib, Querying a summary of database, J Intell Inf Syst, 26, 2006, pp. 59–73
18. S. Wang, B. Ooi, A. Tung, L. Xu, Efficient Skyline query processing on P2P Networks, Proc. ICDE'07, 2007
19. D. Zinn, Skyline Queries in P2P Systems, Master's Thesis, TU Ilmenau, Germany, 2005