
Analyse formelle de concept pour le routage des requêtes dans les systèmes pair-à-pair

Taoufik YEFERNY* – Khedija AROUR**

* *Faculté des Sciences de Tunis
Campus Universitaire, Tunis 1060, Tunisie
yeferny.taoufik@gmail.com*

** *Institut National des Sciences Appliquées et de Technologie de Tunis, 1080 Tunisie
Khedija.arour@issatm.rnu.tn*

MOTS-CLÉS : P2P, routage des requêtes, Apprentissage, LPS

KEYWORDS: P2P, Query routing, Learning, LPS

1. Introduction

Les systèmes pair-à-pair (P2P) ont remporté ces dernières années un succès rapide auprès du grand public et des professionnels car ils offrent la possibilité aux utilisateurs de partager et d'accéder à des ressources diverses, distribuées à large échelle. De nombreuses recherches concernant la sélection des meilleurs pairs contenant les données appropriées à une requête, ont émergé et constituent un axe de recherche très actif. Nous présentons dans ce papier une amélioration de notre algorithme de routage des requêtes par apprentissage LearningPeerSelection (LPS) présenté dans CORIA 2009 et une étude expérimentale avancée.

2. Limites de l'algorithme *LPS*

Bien que notre algorithme *LPS* a donné des bons résultats par rapport aux autres algorithmes, il possède deux problèmes majeurs qui peuvent dégrader ces performances :

– La stratégie de mise à jour des bases de connaissances est une stratégie statique qui consiste à régénérer la base de connaissances de chaque pair à partir de l'historique de toutes les requêtes émises. Cette stratégie est très coûteuse en terme de temps si le nombre d'objets (Requêtes) et d'attributs (termes dans le contexte $C1$ ou pairs dans le contexte $C2$) devient important ce qui dégrade les performances de l'algorithme de génération de concepts formels et pose des problèmes de passage à l'échelle de cette approche.

– La complexité de l'algorithme de sélection des pairs LPS est de $O(|E_2| \times |E_1|)$, avec $|E_i|$ dénote la cardinalité de l'ensemble E_i . Le problème majeur de cette solution est l'exploration importante d'un espace de recherche alors que la contrainte du seuil de nombre maximum des pairs à sélectionner n'a pas été exploitée.

3. Amélioration de l'algorithme LPS

3.1. Stratégie incrémentale de mise à jour de la base de connaissances

L'idée sous-jacente à notre proposition consiste à trouver une nouvelle stratégie de mise à jour de la base de connaissances permettant de prendre en compte les informations sur nouvelles requêtes sans dégrader les performances de l'algorithme de génération des concepts formels. Pour contourner ce problème, nous proposons de définir une stratégie incrémentale qui génère une base B_+ à partir de l'historique des nouvelles requêtes (les requêtes émises après la dernière opération de mise à jour). Par la suite, nous construisons la nouvelle base de connaissances en faisant l'union de B_+ et l'ancienne base. Avec cette stratégie, le nombre d'objets et d'attributs nécessaires pour la construction de la base B_+ est largement inférieur à celui de la stratégie statique. Par conséquent, le temps de mise à jour sera inférieur que celui de la stratégie statique.

3.2. Nouvelle version de l'algorithme LPS

Pour réduire l'espace de recherche, nous proposons de considérer la contrainte du seuil de nombre maximum des pairs à sélectionner P_{max} dans le processus de construction des pairs sélectionnés en parallèle avec la sélection des concepts similaires à Q . Par conséquent, la complexité de la nouvelle version de notre algorithme est de $O(P_{max} \times (|E_1| + |E_2|))$ qui est inférieure à celle de l'algorithme LPS .

4. Impact de la stratégie de maintenance sur la qualité de routage

Les expérimentations que nous avons mené montrent que les deux stratégies donnent les mêmes performances en terme de rappel et du nombre de messages. Il y a une légère différence due à la sélection aléatoire de quelques pairs.