Calcul de l'intersection entre listes triées à base de sauts

Lougmiri Zekri*, Faïza Manseur**, Oussama Belhadj***

Université d'Oran1 Ahmed Ben Bella,
Faculté des Sciences Exactes, Département d'informatique
BP 1524, El-M'naouer, 31000, Oran, algérie
*lougmiri@gmail.com, **faiza_inf@hotmail.fr, ***vipvip215@gmail.com

Résumé. Dans les moteurs de recherche, la réalisation des index passe par le calcul de l'intersection entre les documents alors que l'évaluation des requêtes est le fruit de l'intersection entre ces requêtes et les index. Ce problème de calcul de l'intersection entre des ensembles triés a suscité beaucoup d'intention depuis 1971, date d'apparition du premier algorithme. L'optimisation du nombre de comparaisons et des temps de calcul est le principal objectif que les algorithmes doivent atteindre. Dans ce contexte, nous présentons un nouvel algorithme de type diviser-pour-régner pour le calcul de l'intersection entre des listes ordonnées. Le prétraitement sur les listes est présenté en détail. Il permet à notre algorithme d'éviter de comparer des parties qui ne seront pas forcément partagées par ces listes. Les expérimentations menées montrent que notre solution est performante.

1 Introduction

Le calcul de l'intersection entre listes ordonnées est à la base des processus de composition des index et l'évaluation de requêtes, principalement les requêtes conjonctives dans les moteurs de recherche. L'importance de ce calcul revient au fait que les systèmes informatiques actuels sont larges où la production des documents augmente en volume chaque jour. Le traitement des requêtes doit être aussi rapide que possible d'autant plus que le flux des messages circulant dans le système est important. Les moteurs de recherche doivent gérer des listes inversées immenses afin de répondre en quelques millisecondes à des milliers de requêtes. Le calcul de l'intersection date de 1971 avec les travaux de Hwang et Lin (1971, 1972). Les auteurs ont proposé et étudié un algorithme dit linéaire de fusion entre deux tableaux. Cet algorithme est dit aussi naïf dans la mesure où il scanne de façon séquentielle deux tableaux triés dans l'ordre croissant. Les tests se font sans index et sans la définition de mécanismes pouvant éviter des tests inutiles. Depuis, plusieurs travaux ont vu le jour. Certains travaux ont amélioré le temps moyen de l'intersection et d'autres ont implémenté des procédés qui exploitent les propriétés hard des machines sur lesquelles ils s'exécutent. Le calcul parallèle offert par les nouvelles technologies est aussi un moyen efficace pour calculer cette intersection.

La réduction du nombre de comparaisons et l'accélération de ces comparaisons sont le cheval de bataille de tous ces algorithmes et ces mécanismes. Considérons deux listes triées A et B de longueur n, et supposons que l'on veut localiser les éléments de la liste A dans la liste