

Apport des agents mobiles à l'optimisation de requêtes réparties à grande échelle¹

Franck Morvan, Abdelkader Hameurlain

*Institut de Recherche en Informatique de Toulouse – Université Paul Sabatier
118, route de Narbonne
F-31062 Toulouse Cedex*

email : {morvan, hameur}@irit.fr

1. Introduction

Dans un contexte de bases de données réparties à grande échelle, l'exécution d'un plan engendré par un optimiseur classique peut présenter des mauvaises performances à cause de la centralisation des décisions prises par l'optimiseur, de l'imprécision des estimations, et de l'indisponibilité des ressources.

Les statistiques stockées dans les catalogues système, relatives aux données de base, sont sujettes principalement à l'obsolescence. En effet, dans un réseau à grande échelle, il est très difficile de maintenir continuellement à jour les statistiques concernant les relations de base distantes. Les estimations qui reposent sur celles-ci, de la sélectivité des prédictats (de jointure et de sélection) à la taille des résultats intermédiaires, ne peuvent alors être que plus imprécises. De plus, les estimations relatives à la disponibilité des ressources en termes de la charge des processeurs, de la taille mémoire disponible, et de la bande passante en E/S et en réseau ne peuvent être que fausses.

Les problèmes d'optimisation dus aux imprécisions des estimations et à l'indisponibilité de données ont été intensivement et largement étudiés dans un environnement parallèle et réparti [Hel 00, Kab 98, Kos 00] en considérant uniquement les modèles d'exécution distribuée classiques tels que l'échange de messages et l'appel de procédure distante. De plus, nous pouvons constater que toutes les méthodes d'optimisation dynamique de requêtes proposées à ce jour sont *centralisées*. Elles dépendent du nœud ou du site maître responsable de l'optimisation. Ainsi, quel que soit l'environnement (centralisé, parallèle ou réparti) les méthodes d'optimisation dynamique sont supervisées par un processus maître qui contrôle tous les processus participant à l'optimisation et/ou à la réoptimisation dynamique [Kab 98] pour effectuer toutes les adaptations nécessaires. Les processus réalisant les opérations en cours d'exécution ou en attente de libération de ressources sont donc *entièrement contrôlés* par l'optimiseur qui présente alors un *goulet d'étranglement*. Il engendre, en plus, un échange de messages relativement important sur un réseau à faible débit et à forte latence.

Dans un réseau à grande distance, il est nécessaire de rendre autonome et adaptable l'exécution d'une requête pour éviter le goulet d'étranglement et réduire le volume de données transférées et le nombre de messages de contrôle. Une approche pour résoudre le problème est d'utiliser le modèle de programmation à base *d'agents mobiles* [Ber 02, Fug 98]. Un agent mobile est une entité logicielle autonome et adaptable, capable de se déplacer dynamiquement (code, données et état d'exécution) pour accéder à des données ou à des ressources distantes. La différence fondamentale avec la migration d'activité classique est principalement l'initiateur du déplacement. Il est important de remarquer qu'actuellement, les plates-formes de programmation à base d'agents mobiles ne fournissent que *des mécanismes de déplacement* (migration), mais n'offrent pas de *politique de décision de déplacement*.

Pour ces raisons multiples, nous étudions l'apport des agents mobiles à l'optimisation de requêtes réparties à grande échelle à deux niveaux (i) dans le processus d'optimisation dynamique en réagissant à la fois aux imprécisions des estimations (i.e. taille des relations intermédiaires, et taille mémoire disponible), et à l'indisponibilité des données demandées, et (ii) dans la décentralisation du contrôle effectué par l'optimiseur.

¹ Une partie de ce travail est soutenue par le programme d'actions intégrées Amadeus 2002 (No. 03763 RE) et est menée en collaboration avec l'équipe « Ingénierie des logiciels concurrents et répartis » de l'IRIT.

2. Modèle d'exécution

Dans l'objectif de concevoir et de développer un modèle d'exécution pour des requêtes réparties mobiles (mobilité logicielle : code et données) [Ham 02], il devient donc opportun de rendre, d'abord, *autonome* chaque agent réalisant un opérateur relationnel (e.g. jointure), conscient de son *environnement*, et capable de décider *de l'endroit où* il se déplace pour continuer son exécution sur le site destination. Cet opérateur mobile constitue la *brique de base* pour une exécution autonome et adaptable d'une requête. Dans cette perspective, nous avons défini une politique de déplacement des agents mobiles [Arc 02] issus des opérateurs de requêtes réparties soumises au système en réagissant aux imprécisions des estimations. Avec la mobilité logicielle, les algorithmes de jointure directe et à base de semi-jointure ont été étendus afin de décentraliser la décision et le contrôle. Ce n'est plus l'optimiseur qui choisit où seront exécutées les jointures, mais c'est la jointure elle-même, qui prend la décision de se déplacer ou non. En effet, l'agent exécutant une jointure s'adapte à l'évolution de l'environnement système et réagit aux imprécisions des estimations dues au profil de l'application (e.g. taille des relations intermédiaires) calculées à la compilation et en tenant compte des caractéristiques des ressources systèmes (e.g. bande passante réseau).

L'évaluation des performances, basée sur un modèle de simulation, montre les bénéfices obtenus grâce à la réactivité des agents mobiles aux erreurs d'estimation aux niveaux du facteur de sélectivité FS et de la taille des relations temporaires. Les gains obtenus pour une jointure directe mobile sont très encourageants. Ils sont de 3% à 60% lorsque l'erreur d'estimation du FS est supérieure à 40% et de 11% à 400% lorsque l'erreur d'estimation d'une relation temporaire est supérieure à 30%.

3. Travaux en cours

Les résultats obtenus relatifs à l'apport des agents mobiles à l'optimisation des opérateurs relationnels sont considérés comme des optimisations locales. Ils ne permettent pas de tirer des conclusions quant à leur traduction systématique pour une optimisation globale au niveau d'une requête entière. Ce travail nécessite donc d'être enrichi et étendu par :

1- l'étude de l'apport des agents mobiles à deux niveaux :

- i) dans l'optimisation dynamique des requêtes entières, notamment, en définissant une stratégie pour les liens de poursuite qui acheminent les relations temporaires vers les opérateurs qui se sont déplacés. Cette stratégie devrait éviter la recopie des messages pour ne pas détériorer les performances en terme de temps de réponse.
- ii) dans le processus de décentralisation du contrôle effectué par l'optimiseur évitant ainsi un goulet d'étranglement;

2- l'évaluation des surcoûts dus au contrôle décentralisé et la détermination de son impact sur les performances.

4. Références

- [Arc 02] J.-P. ARCANGELI, A. HAMEURLAIN, F. MIGEON, F. MORVAN « An Adaptive Hash Join Algorithm using Mobile Agents », in : 3rd workshop on Java and Databases : Persistence Options, JaDa 02, 7-10 Oct. 2002, Erfurt, Germany.
- [Ber 02] G. BERNARD, L. ISMAIL, « Apport des agents mobiles à l'exécution répartie », dans: Technique et science informatiques, Hermès, Paris, 2002, Vol. 21, No 6/2002, pp. 771-796.
- [Fug 98] A. FUGGETTA, G. P. PICCO, G. VIGNA, « Understanding Code Mobility », in : IEEE Transactions on Software Engineering 24(5), 1998, pp. 342-361.
- [Ham 02] A. HAMEURLAIN, F. MORVAN, P. TOMSICH, R.M. BRUCKNER, H. KOSCH, P. BREZANY, « Mobile Query Optimization based on Agent - Technology for Distributed Data Warehouse and OLAP Applications », in : First Intl. Workshop on Very Large Data Warehouses, 2 - 3 sept. 2002, Aix-en-Provence (France), pp. 795-799.
- [Hel 00] J.M. HELLERSTEIN, M.J. FRANKLIN, « Adaptive Query Processing: Technology in Evolution », in : Bulletin of Technical Committee on Data Eng., Vol. 23, No. 2, IEEE Press, CA, 2000, pp. 7-18.
- [Kab 98] N. KABRA, D.- J. DEWITT, « Efficient Mid-Query Re-Optimization of Sub-Optimal Query Execution Plans », in : Proc. of the ACM SIGMOD int'l. Conf. on Management of Data, Vol. 27, ACM Press, NY, 1998, pp. 106-117.
- [Kos 00] D. KOSSMANN, « The State of the Art in Distributed Query Processing », in : ACM Computing Surveys, Vol.24, No. 4, ACM Press, NY, 2000, pp. 422-469.