
TRANSMOBI : INTERGICIEL POUR LA GESTION DE TRANSACTIONS MOBILES

Patricia Serrano-Alvarado, Claudia L. Roncancio, Michel E. Adiba

Laboratoire LSR-IMAG
BP 72, 38400 Saint Martin d'Hères, France

Résumé — La communication dans les environnements mobiles est caractérisée par des débits variables ainsi que par des déconnexions fréquentes. Ces caractéristiques compliquent la gestion de données et restreignent la qualité de service fournie par les applications. Dans ce contexte, nous relevons le besoin d'indépendance des unités mobiles en accès aux données, pour surmonter les problèmes dus à la variabilité. Nous introduisons ici TransMobi, un intergiciel dédié au support de l'exécution de transactions mobiles. TransMobi réagit à des événements de changement d'environnement en permettant d'adapter l'exécution d'une transaction mobile (AMT) afin d'augmenter ses chances d'aboutir en accord avec les objectifs de l'utilisateur.

1. Introduction et Motivation

L'arrivée de la technologie de communication 3G UMTS, la prolifération du Wi-Fi (WLAN IEEE 802.11b) et l'évolution des caractéristiques des unités mobiles (e.g. capacité de mémoire et de calcul) contribuent à la croissance de la popularité et la diversité des applications mobiles. Cette évolution défie l'actuelle technologie d'accès et de gestion de bases de données car les approches traditionnelles ne sont pas toujours directement applicables au contexte mobile.

A l'heure actuelle, dans les applications de bases de données les plus courantes, les unités mobiles (UM) jouent souvent un rôle de client. Les clients mobiles accèdent aux services fournis par des serveurs localisés dans le réseau fixe. Par exemple, dans un aéroport, un client mobile peut télécharger les horaires des départs/arrivés des vols. Les UM peuvent ainsi offrir déjà un certain nombre d'applications. Cependant, il semble intéressant d'aller plus loin en permettant aux utilisateurs mobiles d'effectuer des tâches plus complexes, avec une gestion des données plus riche et une certaine fiabilité. Cela conduit à faire évoluer le rôle des « clients mobiles ». Le stockage d'une base de données en local leur permet déjà de participer activement à l'accès aux données, même si cette participation est encore restreinte.

Dans ce contexte, nous nous intéressons à l'exécution de « transactions mobiles » (TM) qui sont pour nous des transactions où au moins une UM participe à l'exécution [8]. L'objectif principal de notre travail est de fournir des transactions fiables et cela à tout moment, malgré les caractéristiques des environnements mobiles. En effet, la variabilité de bande passante, le coût de communication et les déconnexions fréquentes (entre autres) rendent l'implantation des transactions traditionnelles inappropriée. Nous proposons TransMobi¹ [7], un intergiciel dédié au support de l'exécution de transactions mobiles. TransMobi réagit à des événements de changement de l'environnement (UM, réseaux mobile) en permettant d'adapter l'exécution d'une transaction mobile afin d'augmenter ses chances d'aboutir en accord avec les objectifs de l'utilisateur.

Ainsi, nous cherchons à contribuer à l'indépendance des applications d'une UM en leur facilitant la gestion de données cohérentes en dépit du contexte mobile.

Notre travail se déroule dans le cadre du projet NODS [2] (*Network Open Database Services*), qui cherche à définir une infrastructure logicielle répartie ouverte pour de systèmes d'information à différentes échelles.

2. TransMobi

Nous considérons un réseau cellulaire composé d'unités mobiles et d'unités fixes (UF), où chaque cellule est contrôlée par une UF appelée station base (SB). Dans ce contexte, nous distinguons quatre *modèles d'exécution* pour les transactions mobiles : (1) la TM est exécutée complètement sur des UF, (2) l'exécution de la TM est distribuée entre une UM et des UF, (3) l'exécution est distribuée entre plusieurs UM, et (4) la TM est exécutée complètement sur l'UM.

Les principales fonctions de l'intergiciel TransMobi sont la coordination de l'exécution de transactions mobiles (AMT), l'adaptation de leur exécution aux variations de l'environnement mobile et le support de déconnexions des UM. Pour cela, TransMobi support les quatre modèles d'exécution cités.

TransMobi se situe entre l'application mobile et le système de gestion de base de données (SGBD) (voir Figure 1). Son architecture suit le schéma trois-tiers (client/agent/serveur), avec le client sur une UM, l'agent sur une SB (ou une UF pouvant communiquer avec des UM) et le serveur sur une UF. L'agent peut gérer plusieurs clients et le serveur plusieurs agents. Les TM peuvent être demandées par des UM ou UF.

Afin d'obtenir l'adaptabilité, nous proposons un modèle de transaction mobile adaptable (AMT) [7]. Les AMT sont proches des transactions DOM [1] et Flex [4]. L'idée générale est que TransMobi intercepte les appels aux transactions (définies comme des AMT) des programmes d'application (AP). D'après l'information décrite dans l'AMT et l'état de l'environnement mobile courant, TransMobi décide de la façon d'exécuter la transaction. Ensuite, TransMobi perçoit des événements issus des variations de l'environnement mobile et prends les mesures nécessaires pour adapter l'exécution des transactions.

¹ Anciennement appelé MTS

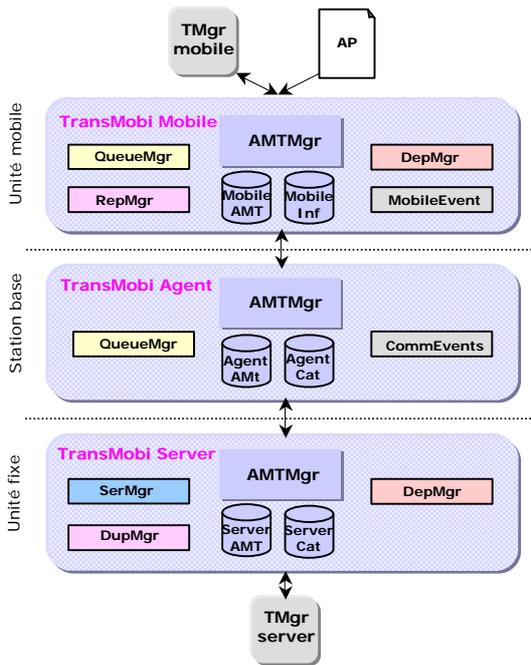


Figure 1 — Décomposition fonctionnelle de TransMobi.

Nous considérons l'existence d'un (ou plusieurs) serveurs de bases de données localisés sur le réseau fixe ainsi que d'une BD sur l'UM (optionnel). L'interaction entre TransMobi et les SGBD sous-jacents est faite à travers des gestionnaires de transactions (TMGr). Cette interaction est transparente pour les TMGr, TransMobi n'est qu'un client demandant l'exécution de transactions. Le SGBD sur l'UM exécute des transactions au profit d'elle-même mais aussi d'autres UM ou UF.

TransMobi est composé de plusieurs modules fonctionnels qui aident à gérer des transactions mobiles adaptables. Il utilise des structures de données stockées sur les trois-tiers (MobileAMT, MobileInf, AgentCat...). TransMobi implante un service de duplication de données [3] (RepMgr) pour les applications qui en ont besoin. La sérialisabilité globale, lors de l'exécution de transactions mobiles distribuées (type 2 et 3) est gérée par SerMgr. Les déconnexions des UM sont gérées par des gestionnaires de queues de messages persistantes (QueueMgr).

Suite à la caractérisation des environnements mobiles, nous avons défini les événements reflétant la variation et affectant l'exécution des transactions : *e-connexion*, *e-disconnexion*, *e-hand-off*, *e-bandwidth-rate-changes*, *e-communication-cost-changes*, *e-available batterie-changes*, *e-available-cache-changes*, *e-available-persistent-memory-changes*. Un service d'événements (MobileEvents, CommEvents) prend en charge leur gestion.

3. Mise en œuvre

L'architecture logicielle de TransMobi est modulaire afin de pouvoir facilement construire un logiciel à base de composants. Nous cherchons à offrir une certaine adaptabilité au niveau logiciel pour permettre de

déployer TransMobi à la demande, avec seulement les modules nécessaires. Cette facilité est exploitée notamment pour les modèles d'exécution car toutes les applications n'utilisent pas les quatre modèles proposés. Il est ainsi possible d'obtenir des empreintes logicielle de taille réduite et minimiser le surcoût à l'exécution dû à d'éventuels objets créés et non utilisés.

A présent, nous avons un premier prototype de TransMobi réalisé en Java. L'UM utilisée est un iPaq H3850 de Compaq avec une version bêta de la machine virtuelle Personal Java. L'infrastructure réseau sans fil est fournie par un WLAN à 11 mbit/s avec 3 points d'accès (SB) qui permettent de couvrir une grande partie de notre bâtiment.

4. Autres travaux et perspectives

L'état de l'art réalisé au début de nos travaux [8,9], nous a montré que les propositions concernant les TM (entre autres [5,6,10]) sont souvent destinées à des contextes particuliers. Nous notons également que la variation de l'environnement mobile est très peu prise en compte et que les modèles d'exécution adoptés sont assez limités. En plus, à notre connaissance, la majorité des travaux n'ont pas été mis en œuvre. Dans notre travail, nous cherchons à offrir un intergiciel facile à intégrer et dont l'utilisation soit transparente pour les SGBD sous-jacents. Par ailleurs, une des originalités de TransMobi est l'adaptabilité pour l'exécution des TM prenant en compte les variations de l'environnement.

La définition de TransMobi est en cours d'achèvement. Les travaux pour compléter le prototype sont également en cours et nous commençons la l'évaluation des performances. Cette évaluation sera orientée vers l'obtention d'indices de performance lors de l'exécution de transactions mobiles avec les quatre modèles d'exécution adoptés.

Bibliographie

- [1] A. Buchmann, M. Tamer Ozsu, et al. A Transaction Model for Active Distributed Object Systems. In *Database Transaction Models for Advanced Applications*, 1992.
- [2] C. Collet. The NODS project: Network Open Database Services. In *ECOOOP 2000*.
- [3] S. Drapeau, C. Roncancio, P. Déchamboux. RS2.7: An Adaptable Replication Framework. In *BDA*, 2002.
- [4] A. Elmagarmid, Y. Leu, and Risinkiewics. A Multidatabase Transaction Model for INTERBASE. In *VLDB*, 1990.
- [5] J. N. Gray, et al. The Dangers of Replication and a Solution. In *SIGMOD*, 1996.
- [6] E. Pitoura and B. Bhargava. Maintaining Consistency of Data in Mobile Distributed Environment. In *TKDE*, 1995.
- [7] P. Serrano-Alvarado. Defining an Adaptable Mobile Transaction Service. In *LNCS 2490*, 2002.
- [8] P. Serrano-Alvarado, C. Roncancio, M. Adiba. Analyzing Mobile Transactions Support for DBMS. In *MDDS*, 2001.
- [9] P. Serrano-Alvarado, C. Roncancio, M. Adiba. Mobile Transaction Supports for DBMS. In *BDA*, 2001.
- [10] G. D. Walborn and P. K. Chiranthi. Transaction Processing in PRO-MOTION. In *SAC*, 1999.