

Gestion des Transactions en environnement Mobile

N. Nouali-Taboudjemat¹, N. Badache²

1Laboratoire Réseaux et Systèmes Répartis
CERIST

Rue des trois frères Aissiou, Ben Aknoun, Alger, Algérie

E-mail : nnouali@wissal.dz

2 Laboratoire Des Systèmes Informatiques,
CERIST/ Institut d'Informatique, USTHB

E-mail : badache@ist.cerist.dz

1. Introduction

Les ordinateurs deviennent de plus en plus omniprésents dans notre vie quotidienne à travers un nombre d'applications en constante augmentation. De ce fait, la demande en ordinateurs portables permettant de se connecter en dehors de chez soi ou de son lieu de travail est, elle aussi, de plus en plus importante. D'autre part, l'évolution rapide de la technologie dans les domaines de la communication cellulaire, des réseaux locaux sans fil et des services dispensés à travers les satellites de télécommunication, permettra à terme à des usagers munis d'unités de calcul portables d'accéder à l'information n'importe où et n'importe quand. Cette liberté de mouvement est d'accès à l'information que l'utilisateur acquiert chaque jour un peu plus, est une raison suffisante pour encourager le développement de systèmes d'information pour ce nouvel environnement de calcul appelé **environnement sans fil et mobile** ou **nomade** (*wireless and mobile environment*).

Des domaines d'application de ce type de systèmes tels que les compagnies de transport, les représentants de commerce, les équipes de maintenance, la télé-médecine, le télé-enseignement, les services d'information de types accès aux pages jaunes et spectacles, la messagerie électronique et les applications

coopératives sont déjà en voie de développement, voire même d'utilisation.

L'environnement mobile ou nomade (*nomadic*) est caractérisé par de fréquentes déconnexions, des limitations significatives du débit de transfert de l'information et des unités portables pourvues de ressources très contraignantes. En effet, les sources d'énergie nécessaires au fonctionnement des unités portables, leur capacité disque et mémoire et la vitesse du processeur limitées ainsi que les changements fréquents de localisation sont les principales caractéristiques de ces dernières.

L'accès par les utilisateurs aux masses d'informations et données qui lui sont offertes va inévitablement impliquer des transactions. Les transactions sont utilisées depuis les années soixante dix comme un outil permettant un accès préservant la cohérence des données, d'abord dans les applications de base de données classiques (applications bancaires et de réservation de place d'avion, etc.), puis dans les applications avancées (CAD, bases de données orientées objet, etc.).

Cependant, les transactions doivent faire face à de nouveaux défis induits par les communications non filaires et la mobilité. Dans ce papier, nous examinons ces nouveaux défis et leur impact sur les transactions (modèles et propriétés). Nous présentons et « discutons » les résultats des travaux de recherche menés dans ce domaine.

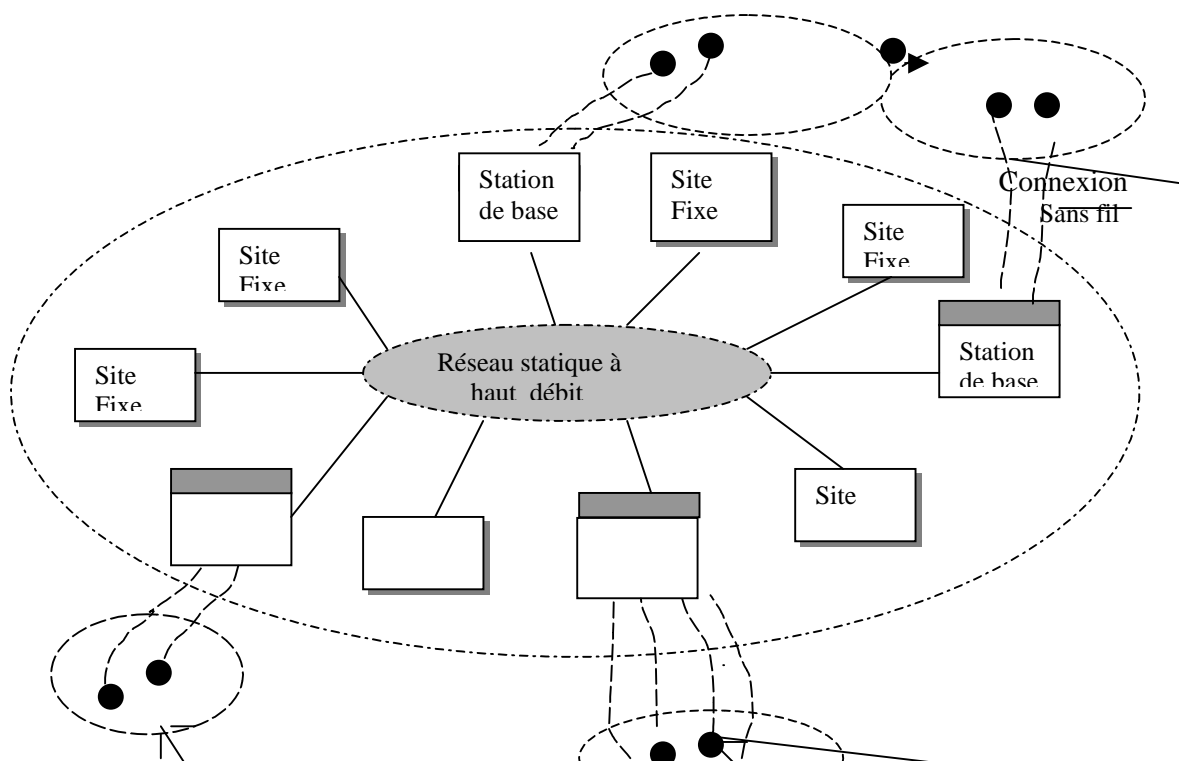
2. Architecture d'un système mobile :

La figure 1 montre le modèle d'architecture, largement accepté [DUN 95, IMI 94, BAD 98], d'un système supportant un environnement de calcul mobile. Un ensemble d'ordinateurs (PCs, stations de travail, etc.) sont interconnectés par un réseau de communication filaire classique à haut débit. Parmi ces ordinateurs certains sont des sites (ou hôtes) fixes (*Fixed Hosts*,

FH). D'autres sont des stations de base, SB (*Base Stations, BS*) ou station de support de base (*Mobile Support Station, MSS*) munies d'une interface de communication sans fil, leur permettant de communiquer avec les sites mobiles. Les sites ou hôtes mobiles (*Mobile Hosts, MH*) ou unités mobiles, UM (*Mobile Units, MU*) sont connectées aux SBs à travers des canaux sans fil. Les sites fixes ne sont pas équipés pour communiquer directement avec les UMs. Les SBs fournissent une passerelle de communication entre la partie filaire et la partie non filaire du réseau. Les stations de base ne sont pas limitées à la fonction de routage ou de communication mais peuvent supporter des applications tels qu'un serveur de base de données (voir figure 1).

Les UMs peuvent être de configurations diverses : avec ou sans disque, des capacités de mémorisation et de calcul plus ou moins modestes et sont alimentées par des sources d'énergie autonomes (batteries). Elles peuvent se déplacer librement dans une zone géographique limitée, qui pour une meilleure réutilisation des fréquences, est elle-même subdivisée en *cellules* de plus petite taille. Chaque cellule est gérée par une SB particulière.

Une UM peut se déplacer d'une cellule à l'autre, mais ne peut être connectée à un instant donné qu'à une seule station de base. Le processus de passage d'une cellule à l'autre est assuré par un protocole appelé *Handoff*. Ce processus est transparent aux UMs. Il est responsable du transfert des informations d'état, relatives au calcul mobile en cours, vers la station de base de la nouvelle cellule [PIT 94b, DUN 97, WAL 97, BAD 98].



Unités mobiles

Cellules de communication sans fil

Figure 1 : Architecture d'un système mobile

3. Caractéristiques d'un système mobile :

La **mobilité** des utilisateurs et donc des **unités portables**, les **caractéristiques physiques** de ces dernières ainsi que les **communications sans fil** induisent des problèmes que nous résumons dans ce paragraphe [SAT 96, FOR 94, IMI 94, ALO 93].

La **mobilité** fait que la localisation des UMS et donc de leur point d'attache au réseau fixe change en fonction de leur déplacement d'une cellule à l'autre. Les conséquences de cette mobilité sont nombreuses :

- La configuration d'un système comportant des unités mobiles n'est pas statique. Ceci implique que la conception d'algorithmes distribués ne peut plus être basée sur une topologie fixe (anneau, arbre, ...).

- Le coût induit par la localisation des unités mobiles s'ajoute au coût de chaque communication les impliquant.

- L'hétérogénéité en terme de connectivité (*connectivity*) qui varie sensiblement en fonction de la position géographique (par exemple, à l'extérieur et à l'intérieur d'un bâtiment la bande passante n'est pas la même), et aussi en termes de ressources disponibles aux niveaux des UMs (mémoire plus ou moins importante, écran de dimension variable etc.). La mobilité soulève aussi des problèmes très importants liés à la sécurité et à l'authentification.

Le médium de communication non filaire se caractérise par :

- Une connexion faible et intermittente à cause du coût des communication de ce type et de la largeur de bande faible. Ainsi, alors qu'en environnement filaire d'énormes progrès ont été réalisés en terme de largeur de bande (la technologie actuelle offre 10 Mbps pour Ethernet, 100 Mbps pour FDDI et 155 Mbps pour ATM), les produits pour les communications non filaires atteignent 19 Kbps seulement pour les ondes radio et 9 à 14 Kbps pour la téléphonie cellulaire. La largeur de bande typique d'un réseau local non filaire (Wireless Local Area Network) se situe entre 250 Kbps et 2 Mbps et pourrait atteindre la valeur de 10 Mbps. Du fait que la bande disponible est partagée entre les utilisateurs appartenant à la même cellule, la largeur de bande exploitable par un utilisateur est encore plus faible. Concernant la transmission radio, le taux d'erreur est si élevé que la bande effective ne dépasse pas les 10 Kbps. La bande passante est actuellement une ressource rare et malgré les progrès de la technologie, il n'est pas faut de supposer qu'elle restera une limitation importante et une contrainte de performance dont il faudra tenir compte dans la conception d'un système mobile.

- Une déconnexion facile. La déconnexion d'une unité mobile du reste du réseau peut être volontaire ou involontaire, prévisibles ou soudaines, de courtes ou de longue durée.

- Un support physique de la diffusion. Un canal de diffusion très large partant de la station de base vers les unités mobiles est offert. C'est un avantage à exploiter dans le calcul mobile.

La portabilité des unités mobiles a rendu possible leur mobilité. Cependant, cette mobilité est obtenue au prix de limitations dans les ressources de calcul :

- Les unités mobiles doivent être légères et de petites dimensions afin d'être facilement transportables. De ce fait, elles sont moins équipées (en mémoire, taille de l'écran et de capacité du disque) que les unités statiques. Ceci conduit à une asymétrie entre les deux types de sites existants sur un même réseau ; un autre facteur à ne pas négliger dans la conception des systèmes de calcul mobile.

- Les unités mobiles comptent sur des batteries à capacité limitée pour leur alimentation. Malgré les progrès technologiques, ce problème ne cessera pas d'exister. Un intérêt doit être porté à la consommation d'énergie à tous les niveaux de conception hardware ou software.

- Les unités mobiles sont sujettes aux accidents, au vol et à la perte. Et sont par conséquent, moins fiables et moins sécurisées que les unités statiques.

4. L'évolution du modèle transactionnel

4.1. Notion de transaction

La notion de transaction a été utilisée pour assurer une gestion de données consistante et des interactions atomiques et isolées entre utilisateurs. Ceci s'est traduit à travers **le modèle transactionnel classique** ou **linéaire** où les transactions étaient modélisées comme une séquence d'opérations de lecture et d'écriture,

ayant une structure plate, un seul début et un seul point de rejet ou de validation. Dans ce modèle les transactions étaient donc caractérisées par les propriétés **ACID (Atomicité, Cohérence ou Consistance, Isolation et Durabilité)** [BAL 91].

Atomicité. La séquence d'actions d'une transaction est indivisible: soit toutes les actions de la transaction sont exécutées et la transaction est **validée** (le système atteint un nouvel état cohérent), soit aucune ne l'est et la transaction est **rejetée** (le système reste dans l'état cohérent initial). C'est le principe du tout ou rien.

Cohérence ou Consistance. L'exécution d'une transaction dans un environnement sans concurrence et sans pannes ne doit pas introduire d'incohérences.

Isolation. Les actions d'une transaction sont isolées. Le résultat des actions intermédiaires (état temporairement incohérent) est masqué aux autres transactions. Une autre manière de présenter cette propriété est que le résultat d'une exécution parallèle de plusieurs transactions est équivalent à celui d'une exécution séquentielle. On parle alors de **sérialisation** (en théorie la sérialisation est plus restrictive que l'isolation).

Permanence ou Durabilité. Les résultats d'une transaction qui s'est bien terminée ne peuvent pas être détruits ultérieurement par l'occurrence d'une panne.

Si toutes ces propriétés sont vérifiées, on parle de cohérence forte (ou cohérence stricte). Si une des propriétés n'est pas vérifiée la cohérence est dite dans un état dégradé ou faible.

4.2. Les modèles de transaction avancés

La dernière décennie a été marquée par un développement impressionnant des bases de données (bases de données orientées objet, géographiques, multimédia, etc.) et de nouvelles applications multi-utilisateurs (dites

avancées) ont alors intégré le concept de transaction, c'est le cas de la bureautique, de la CAO, du génie logiciel et du travail coopératif.

En vue d'une part, prendre en compte les caractéristiques des applications avancées et d'autre part, s'adapter à la répartition, les systèmes transactionnels ont connu une activité intense au sein de la communauté scientifique. Des **modèles de transactions avancés** (par opposition au modèle classique) ont été proposés. Le **modèle de transaction imbriqué** et sa variante le **modèle multiniveau** sont les extensions les plus importantes apportées au modèle transactionnel [WEI 92, BAL 91]. Le modèle de transaction imbriqué (*Nested Transactions*) a été motivé par le souci de mieux intégrer le parallélisme et les points de reprise à l'intérieur d'une transaction. Dans ce modèle une transaction est découpée en unités plus fines (appelées sous-transactions) pouvant s'exécuter séquentiellement ou en parallèle sur un ou plusieurs sites. Ce modèle présente des propriétés intéressantes; en premier lieu il permet de structurer une transaction de longue durée de vie et/ou manipulant des objets complexes et de grande taille, en unités de reprise indépendantes tout en assurant les propriétés fondamentales de sérialisation et d'atomicité globales. En outre il est bien adapté à la répartition dans la mesure où les sous-transactions peuvent s'exécuter en parallèle sur des machines différentes.

Plusieurs variantes de ce modèle ont été proposées dont les modèles **imbriqués ouverts** et **multiniveaux**. Le modèle imbriqué ouvert se distingue par rapport au modèle imbriqué fermé principalement par une granularité d'isolation réduite. En effet, les mises à jour effectuées par une sous-transaction sont rendues visibles aux autres transactions du système immédiatement après sa validation. Le modèle multiniveau permet de faciliter la manipulation d'objets complexes dans les systèmes où l'accès à un objet nécessite des accès à d'autres objets [WEI 92, SCH 95].

Malgré leurs spécificités, d'autres modèles avancés ont été inspirés par le modèle imbriqué ouvert. Tous ces modèles ont établi un consensus sur le fait de subdiviser une transaction en unités plus réduites, en locurence les sous-transactions. Ceci permet de ramener le support de certaines propriétés au niveau des sous-transactions au lieu de la transaction globale. Ces modèles sont basés sur la réévaluation des propriétés ACID. La notion de cohérence sémantique a été introduite afin de « relâcher » la cohérence et permettre la mise en œuvre d'un contrôle moins contraignant et donc d'accroître les performances du système global. L'effort a surtout porté sur l'exploitation de la connaissance de la *sémantique de l'application* pour mettre en œuvre un contrôle plus 'intelligent' permettant d'avoir une sérialisabilité (isolation) moins stricte. Parmi ces modèles nous citons le **modèle Saga** qui a été conçu spécialement dans le but de pallier aux problèmes que posent les transactions de longue durée de vie [GAR 87], et le **modèle flexible** conçu pour s'adapter aux systèmes multibase de données. Dans les systèmes multibase de données deux types d'application coexistent : les applications locales associées aux systèmes existants, et les applications globales provenant de la multibase de données et accédant à plusieurs de ces systèmes en même temps. D'où les deux types de transactions : locales et globales. Le **modèle flexible** structure une transaction globale en la décomposant en un ensemble de tâches pouvant être réalisées chacune de manière différente. Ce qui a donné naissance aux **alternatives** : plusieurs sous-transactions fonctionnellement équivalentes sont associées à chaque tâche. Pour une même transaction flexible plusieurs chemins d'exécution (alternatives) sont possibles [MED 95].

5. Les effets de la mobilité sur les modèles transactionnels

5.1. Les effets de l'environnement mobile sur les transactions

Alors qu'un modèle de transaction pour le calcul mobile n'a pas encore été élaboré, il semble évident qu'une certaine notion de transaction demeurera nécessaire pour résoudre les problèmes d'accès aux données en environnement sans fil et mobile [PIT 94a]. En effet, les besoins des utilisateurs en termes de validité des données qu'ils lisent, de la prise en compte des mises à jour qu'ils effectuent sur ces données, et d'assurance vis à vis d'une gestion efficace de leurs interactions avec d'autres utilisateurs accédant à ces mêmes données restent les mêmes. **Une transaction mobile est définie comme étant une transaction distribuée dont certaines parties s'exécutent sur des sites statiques et d'autre sur des unités mobiles.**

L'utilisation de réseaux de communication sans fils conduit à des transactions de longue durée, à cause des délais de communication longs favorisés par la faiblesse des bandes passantes et la latence de ces réseaux. La nature même des transactions émises par un utilisateur mobile peut conduire à une longue durée de vie (voir exemple ci-dessous figure 3).

Les déconnexions fréquentes mettent en avant le problème de fiabilité (au lieu de la disponibilité) qui devient une exigence primaire du traitement transactionnel mobile. Les raisons de ces déconnexions sont comme on l'a vu dans les paragraphes précédents soit volontaires motivés par le soucis d'économie (coût de la communication sans fil, batterie limitée, ...), soit involontaire (changement de cellule, panne, vol, ...). Le résultat de ces déconnexions fréquentes est que l'exécution d'une transaction mobile est moins fiable.

Un aspect important caractérisant les transactions mobile est justement leur mobilité dans le réseau. Autrement dit, contrairement aux transactions

classiques, une transaction mobile peut être initiée au niveau d'un site et se terminer dans un autre site. D'où la nécessité de disposer d'un système de gestion capable de contrôler leur mouvement. En effet, une transaction mobile s'exécute d'une manière différente d'une transaction distribuée. La première visite un ensemble de sites en « sautant » de l'un à l'autre, alors que la dernière communique avec un ensemble de sites, et commence et se termine au niveau du même site initiateur. Ceci implique la nécessité de revoir les protocoles de coordination de la terminaison (validation et annulation) de la transaction. De plus, la mobilité, permet aux transactions en cours d'exécution d'accéder à des systèmes hétérogènes, à des données de localisation (probablement imprécises) ou à des données dont la localisation (position) change dynamiquement [PIT 94a, NAR 94a].

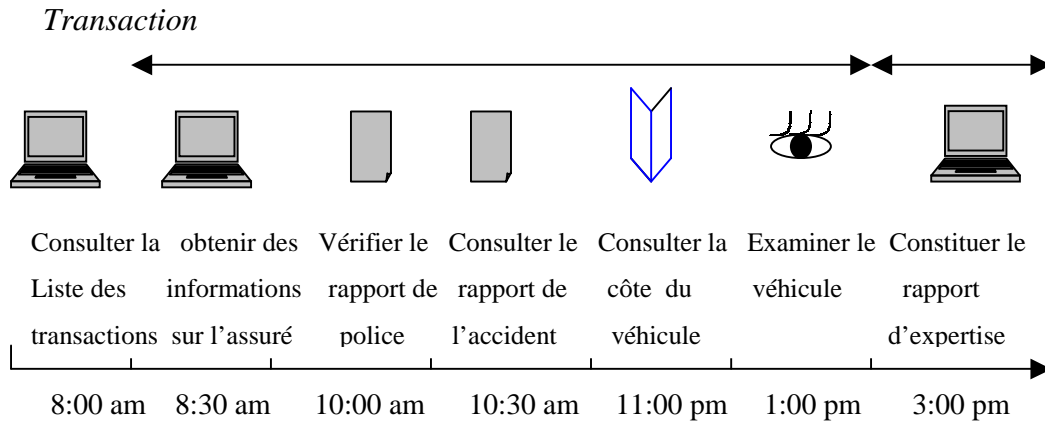
5.2. Exemple de transaction mobile

L'exemple de la figure 3, empreintée à [DUN 96], illustre les exigences d'un modèle de transaction mobile. Il s'agit d'une agence d'assurance automobile dont un employé doit expertiser les dégâts physiques subis par un ensemble de véhicules accidentés et fournir une estimation des coûts nécessaires aux réparations. Afin de réaliser cette évaluation, la personne a besoin d'accéder à des informations concernant le véhicule à examiner, aux rapport de police concernant l'accident, aux informations sur les requêtes émises précédemment par le propriétaire du véhicule en question, et la valeur du véhicule indiquée par le « Livre Bleu » (*Blue Book*). L'objectif de la transaction est de fournir un rapport incluant les informations d'assurance, des informations sur l'accident, le constat des dégâts, l'estimation du coût des réparations, et des recommandations indiquant si oui ou non la compagnie devra payer pour les dégâts, si des poursuites sont à engager contre le client (rompre son contrat), ou contre une tierce partie impliquée dans l'accident). Dans les systèmes actuels, le chef d'agence fait parvenir à l'expert les dossiers à traiter, par écrit ou par un message électronique.

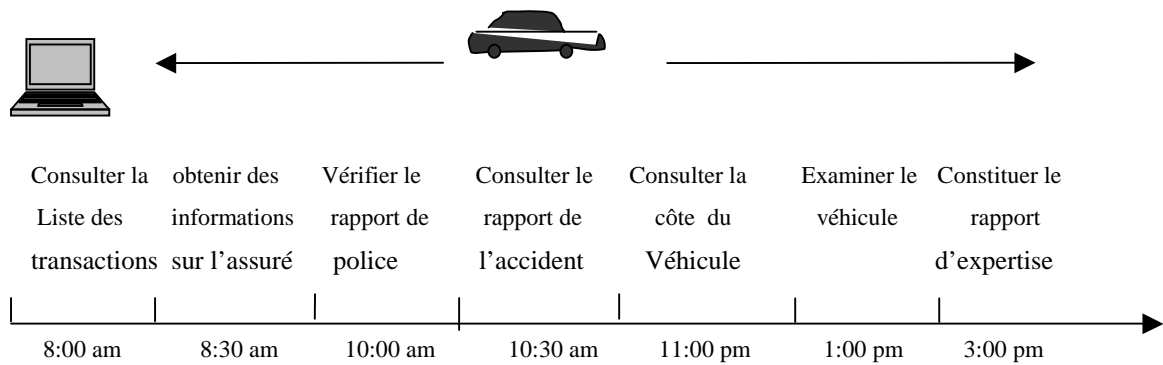
Avant d'aller expertiser le véhicule, ce dernier doit réunir toutes les informations nécessaires à l'accomplissement de son travail. Certaines de ces informations sont obtenues moyennant des appels téléphoniques, d'autres nécessiteront le déplacement de l'expert vers la source d'information, d'autres encore seront recueillies à partir d'un site Internet. Puis, l'expert se déplacera vers le lieu où se trouve le véhicule accidenté. Des notes seront prises sur place sur un support papier. Ce n'est qu'à son retour au bureau que la transaction d'expertise sera effectivement réalisée à partir des données collectées tout au long de la journée. Un rapport sera envoyé au chef d'agence. La majeure partie des tâches de collecte des informations est réalisée hors connexion (*off line*).

Comment peut-on envisager le déroulement de ces tâche en utilisant les transactions mobiles ? Premièrement, la collecte des données sera réalisée de manière connectée (*On line*). Après avoir pris connaissance des dossiers à traiter dans la journées, l'expert décide par quel dossier commencer. Il peut immédiatement, lancer la transaction. Les fonctions que cette dernière doit effectuer sont logiquement les mêmes que celles du système classique, à la différence que la collecte des informations sera prise en charge, dans le nouvel environnement, par la transaction elle même. Des sous-transactions seront générées pour obtenir toutes les données désirées de manière automatique. Ceci est effectué durant le trajet conduisant l'expert, qui se déplace en voiture à la localisation du véhicule endommagé. A l'arrivée il peut retrouver toutes les informations qu'il désire sur son unité portable. Tout en examinant le véhicule, il peut introduire les données qu'il désire dans son ordinateur mobile, qui réalise automatiquement les mises à jours requises sur le SGBD source. Tout les rapports sont archivés à ce moment là. En quittant le véhicule examiné, l'employé aura fini le traitement du dossier. Un autre dossier et donc, une autre transaction peut être entamée sur son chemin vers la localisation du deuxième véhicule à expertiser.

Collecte des données



a) Déroulement d'une transaction classique



b) Déroulement d'une transaction mobile

Figure 3 : Exemple de transaction mobile

L'utilisation de l'unité mobile a plusieurs impacts : d'abord, le traitement des dossiers est plus **rapide et l'efficacité** de l'employé est nettement améliorée ; **aucune saisie des données n'a été réalisée sur papier.**

Puis, le **temps durant lequel la transactions est restée active s'est prolongé.**

D'autres facteurs compliquant la transaction peuvent survenir. Sur son chemin vers le deuxième véhicule, l'expert décide de s'arrêter pour déjeuner et afin d'économiser sa batterie, il éteint son portable. Il le rallume et relance la transaction active au point où il l'a laissée avant son déjeuner. Il est donc possible pour une transaction mobile d'être suspendue avant d'avoir atteint son point de terminaison. Cela peut se produire aussi pour des raisons imprévisibles (une panne de batterie par exemple). Le temps qui s'écoule entre **l'interruption et le relancement de la transaction peut être relativement long.** Les choses se compliqueront si après ce temps l'utilisateur décidait d'annuler la transaction interrompue.

Afin de pouvoir continuer une transaction interrompue, la solution **serait de valider des portions de la transaction dès que possible.** Ces validations précoces peuvent libérer les ressources (par exemple les verrous).

5.3 Les limites des modèles de transactions avancées dans l'environnement mobile

Il est clair que le modèle de transactions atomiques et plates basé sur les propriétés ACID ne peut être utilisé en environnement mobile. En effet, les transactions atomiques sont supposées s'exécuter de façon isolée et indivisible, les empêchant de couper leur exécution ou de divulguer des résultats intermédiaires. Or en environnement mobile, pour des considérations pratiques (voir exemple figure 3) il est nécessaire que les traitements soit partagés entre des unités mobiles et des sites statiques pour les besoins de communication et de calcul.

Le modèle imbriqué où une transaction (père) engendre des transactions (filles), offre plus de flexibilité que le modèle linéaire de transactions atomiques à

structure plate. En effet, le modèle imbriqué supporte les coupures et les pannes partielles. Cependant, les transactions imbriquées ne partagent pas leurs résultats intermédiaires durant leur exécution. La validation se fait de bas (les feuilles) en haut (racine) ou « *bottom-up* ». Et, ce n'est qu'après la validation de la racine que les objets modifiés sont rendus permanents. Ceci signifie, également que l'état du calcul mobile doit être sauvegardé jusqu'à la terminaison complète de la racine (c'est-à-dire de la transaction globale). Supposons que la racine se trouve sur une unité mobile alors que les transactions filles s'exécutent elles sur des sites fixes. Si les sous-transactions ne sauvegardent pas leurs états après leur terminaison, l'état de tout le calcul devra alors être sauvegardé à tout moment sur l'unité mobile malgré ses ressources limitées. Si, par contre, les sous-transactions sauvegardent leurs états, alors l'état du calcul est distribué entre plusieurs nœuds, en fonction du chemin emprunté par l'unité mobile durant ses déplacements, rendant coûteuse l'opération de validation [CHR 93].

Le modèle imbriqué ouvert (*Open-Nested Transaction Model*) tente de « relâcher » quelques restrictions des transactions imbriquées en rendant visibles, à l'extérieur d'une transaction, leurs résultats intermédiaires. Ceci est dû au fait qu'une sous-transaction décide de se valider ou d'avorter indépendamment de ses ancêtres. Ce modèle a été souvent adopté dans le contexte de *systèmes multi-bases de données* (multi-database systems) [GAR 87]. Un environnement de bases de données mobile peut être vu comme un cas particulier de systèmes multi-bases avec des exigences spécifiques. Par exemple, la notion d'autonomie locale en environnement mobile se manifeste dans l'habilité des sites mobiles à continuer leur fonctionnement normal en période de déconnexion [MUM 94]. Deux exigences spécifiques aux transactions mobiles ne peuvent pas être satisfaites par les modèles imbriqués ouverts actuels. La première est la capacité des transactions à partager leurs résultats partiels entre elles *durant leur exécution* , et la deuxième, est

l'aptitude à maintenir une partie du calcul mobile sur un site fixe dans le but de minimiser les délais de communication entre les unités mobiles et leurs stations de bases [CHR 93].

6. Les modèles de transaction pour l'environnement de calcul mobile

Afin de répondre aux nouvelles exigences il est clair que les modèles transactionnels existants ne sont pas adaptés. Des travaux de recherche tentant de répondre à ces besoins sont en cours, mais actuellement les solutions proposées restent assez dispersées. En effet, il n'y a pas encore de consensus sur une approche donnée. La modélisation de transaction pour environnement mobile est abordée dans le cadre des recherches sur les problèmes du traitement mobile de l'information que [KUM 98] a classé en (a) la restructuration des transactions ACID et (b) le développement d'un nouveau modèle de transaction approprié au calcul mobile. Dans la littérature, les solutions proposées se sont généralement basées sur des améliorations apportées aux modèles décrits précédemment (notamment le modèle imbriqué ouvert) mais ne satisfont que partiellement aux caractéristiques exigées. Dans ce qui suit nous survolons les principaux travaux concernant la conception de modèle de transaction mieux adaptés aux nouvelles spécificités imposées par le calcul mobile.

6.1. Le modèle de transactions rapporteuses et co-transactions

Chrysantis [CHR 93] présente un schéma pour la structuration des transactions mobiles en se basant sur le modèle imbriqué-ouvert augmenté des notions de **transactions rapporteuses (*reporting transactions*)** et de **co-transactions (*co-transactions*)**. Une transaction parent est représentée par ces deux types de transactions pouvant s'exécuter soit sur l'unité mobile soit sur la station de base. Une transaction rapporteuse peut partager ses résultats partiels avec

la transaction parent à tout moment et peut valider de manière indépendante. Une co-transaction est une transaction rapporteuse spéciale qui se comporte comme une co-routine. Elle est suspendue par une autre transaction au moment du partage des résultats, puis elle reprendra son exécution à partir de ce point. Alors que la transaction rapporteuse peut, après la diffusion de ses résultats, continuer son exécution et valider. Ces deux types de composants peuvent s'exécuter au niveau de plusieurs sites (UM ou SB) en parallèle ou en série. L'atomicité sémantique peut être assurée grâce à des transactions de compensation. Ce modèle suppose par ailleurs l'existence d'un SGBD global à chaque SB. L'auteur affirme que ce modèle minimise le coût des communication filaires et non filaire.

6.2. Le modèle de clusterisation

Ce modèle est conçu pour un système totalement distribué, et est conçu pour assurer la consistance de la base de donnée. La base de données est divisée en clusters regroupant chacun des données sémantiquement liées ou localisées sur des sites voisins. Une cohérence stricte est exigée pour les données internes à un cluster et des degrés de cohérence sont définis sur des données dupliquées et localisées sur des clusters différents. Le degré de cohérence peut dépendre de la qualité de la liaison entre une unité mobile et sa station de base. Lorsque la connexion est faible, les utilisateurs peuvent tolérer un degré d'inconsistance plus élevé (voir aussi [IMI 93]). Offrir des opérations faibles de lecture et d'écriture, lisant ou écrivant sur une copie disponible localement (dans le cluster) a pour effet d'introduire l'incohérence dans les transactions. D'où la notion de **transactions faibles**. Les mises à jour sont répercutées ultérieurement sur les autres clusters et ne seront validées que si elles n'entrent pas en conflit avec d'autres opérations d'écriture stricte. L'accès se fera globalement (en dehors du cluster) à des données cohérentes à l'aide de **transactions strictes** [PIT 94a, PIT 94c]. En résumé, la transaction mobile sera décomposée en transactions faibles et en transactions

strictes. La décomposition devra se faire sur la base du degré de consistance requise. Les opérations faibles sont autorisées en période de déconnexion. Durant cette phase, les mises à jours éventuelles auront lieu sur l'UM et peuvent devenir inconsistance vis à vis des données se trouvant sur le réseau fixe.

Le concept de **transactions suppléantes (transactions proxies)** a été introduit pour assurer la durabilité des effets du calcul effectué sur l'UM vu leur vulnérabilité. A chaque transaction exécutée sur un site mobile, une transaction duale est définie et exécutée sur la station de base correspondante. Cette transaction suppléante est considérée comme une sous-transaction de la transaction originale. Ainsi, à chaque soumission d'une transaction à l'UM, sa transaction suppléante est soumise à sa SB. Les transactions suppléantes n'incluent que les mises à jour de leur transactions d'origine. Le déplacement de l'unité mobile implique le relogement des transactions suppléantes peut être envisagé. Les transactions suppléantes pourraient également être utilisées pour réaliser des opérations de sauvegardes périodiques (correspondant à la notion de points de reprise) [PIT 94a].

6.3. Le modèle sémantique

Dans [WAL 95] on propose d'exploiter la sémantique des objets (en particulier la commutativité des opérations) afin d'améliorer la gestion des accès concurrents sans pour autant compromettre la propriété de sérialisation (isolation). Le traitement de transactions mobiles est ramené au problème de concurrence et de cohérence du cache. Ce modèle considère une transaction mobile comme étant une transaction de longue durée de vie caractérisée par des délais de transmission longs et des déconnexions non prévisibles. Cette approche l'organisation orientée objet pour découper les objets de grande taille ou complexes en fragments plus petits. Une base de données située sur la partie statique envoie des fragments vers l'UM à sa demande. A la terminaison de la transaction, l'UM retourne les

fragments au serveur. Ces fragments sont regroupés de nouveau par une opération de fusion exécuté au niveau du serveur. Lorsque les fragments peuvent être combinés dans un ordre quelconque ils sont dit « recomposables » (*reordable objects*). Du moment qu'un serveur unique de base de données est présupposé, les propriétés ACID peuvent être maintenues. Les ensembles, les structures de données de type pile ou file d'attente sont des exemples d'objets « fragmentables ».

6.4. Le modèle MTPM (*Multi-database Transaction Processing Manager*)

Cet article a examiné comment une transaction multibase de données pourrait-elle être exécutée à partir d'UMs. Une architecture en couche pouvant supporter ces transactions mobile est proposée. Chaque transaction initiée par un site mobile M, est soumise à SB où elle est exécutée et ensuite le résultat est retourné à M. Le site M peut entre temps évoluer de façon indépendante. Ce modèle est une généralisation des systèmes multibases de données, et de ce fait le maintien des propriétés ACID dépend de ces derniers. Ce modèle offre une approche pratique, cependant, il ignore des problèmes importants dont l'interactivité des transactions nécessitant des entrées émanant de l'utilisateur et produisant des sorties, les transactions impliquant des données sauvegardées sur les sites mobiles, ainsi que la migration des sites mobiles [YEO 94].

6.5. Le modèle de transaction par lot avec cache

Dans [NAR 93] on s'intéresse à la faiblesse de la bande passante et à la latence des communications sans fils dans un environnement mobile. On propose alors une méthode de structuration des transactions pour parer à ces insuffisances et améliorer le temps de réponse.

L'idée est d'exécuter la transaction entière localement sur l'unité mobile moyennant des copies caches des données. Pour sa validation, la transaction entière est envoyée à un serveur de bases de données situé sur un

site fixe pour le reste des opérations. Le serveur négociera alors la validation avec les autres serveurs du système. Un accusé de réception positif est alors envoyé à l'unité mobile si la validation a lieu. Si la transaction venait à être avortée pour avoir utilisé une version périmée de l'une des données manipulées, le serveur envoie la dernière version de la données en question à l'unité mobile. Cette dernière peut alors réexécuter la transaction. Cette méthode est optimiste car elle suppose qu'il y a peu de chances pour que les transactions ne soient pas validées après leur transfert sur le serveur. On peut diminuer les chances d'avoir des conflits et de voir trop de transactions annulées en réduisant la granularité des objets.

6.6. Le modèle Kangourou

Le modèle **Kangourou** [DUN 97] est le premier et, à notre connaissance, le seul modèle transactionnel qui a été présenté avec le souci de modéliser le **mouvement des transactions** (leur mobilité d'une cellule à l'autre). Ce modèle est construit sur une approche multibase de données où la gestion des transactions est toujours réalisée au niveau de la station de base (ou d'un site spécifié sur la partie fixe du réseau). La gestion de la transaction suit son mouvement et se déplace en même temps que le déplacement de l'unité mobile. Chaque transaction Kangourou est divisée en sous-transactions qui sont exécutées indépendamment au niveau des SGBDs ou des SGMBDs (Système de Gestion des Multibases de Données) autonomes situés sur le réseau fixe. En effet, une fois initiée à partir de l'UM, la transaction Kangourou ne peut être exécutée que sur les serveurs se trouvant sur les sites fixes du réseau. Deux modes d'exécution sont permis, l'un assurant une atomicité globale moyennant des transactions compensatrices (*compensating transactions*) pour les sous-transactions. Ces transactions compensatrices permettent de restituer (par annulation des sous-transactions) un état cohérent en cas de défaillance de l'une des sous-transactions. Le mode sans compensation aboutit à la validation/annulation locale des sous-transactions

indépendamment les unes des autres, ne permettant pas d'assurer l'atomicité globale de la transaction.

7. Discussion

En se basant sur les modèles de transaction ci-dessus, ce qui peut être déduit est que la réévaluation des propriétés ACID doit passer par la subdivision de la transaction en sous-transactions. En effet, appliquer ses propriétés à la transaction mobile globale est trop restrictif. On peut retenir la notion **d'atomicité sémantique** pouvant être mise en œuvre grâce à la notion d'opérations ou transactions de **compensation**. On peut aussi définir des degrés de **consistance** (implémentables par le concept de **clusterisation**, d'**opérations strictes** et d'**opérations faibles**). Une atomicité et une isolation faibles peuvent relâcher la propriété de consistance. Une atomicité « relâchée » est en contradiction avec une **isolation** stricte. Cette dernière peut être « relâchée » grâce à la prise en compte de la sémantique des objets (qui permet de réduire la granularité d'**isolation**), des opérations (commutativité des opérations par exemple) et/ou du concept de **transaction rapporteuses** et de **co-transactions**.

La durabilité est difficile à assurer sans la compensation.

Aux propriétés classiques viennent s'ajouter de nouvelles propriétés intrinsèques à la nature même de l'environnement mobile : **la déconnexion** et **le relogement** (*relocation*). **La propriété de relogement** découle directement de la mobilité des utilisateurs. Le relogement des transactions implique le transfert des informations du contexte d'exécution d'une station à une autre pour que l'exécution puisse continuer. Le type et le volume de ces informations dépendent de l'application et affectent la performance du système. Cette propriété constitue un facteur supplémentaire dont il faudra tenir compte dans les nouvelles solutions. Le relogement des transactions et leur

aptitude à s'exécuter sur des sites différents sont deux facteurs qui militent, eux aussi, en faveur du relâchement de la propriété d'atomicité. La déconnexion a largement été traitée dans les modèles présentés en assurant une autonomie d'opération aux unités mobiles. Ceci ne peut être possible sans le « relâchement » de la cohérence et de l'atomicité.

Toutes approches confondues, chacun des modèles proposés a essayé de s'adapter à une (parfois plus) des caractéristiques des transactions mobiles. Le modèle de transactions rapporteuses et co-transactions s'est fixé comme but de remédier au manque d'interactivité dans le modèle imbriqué ouvert et l'a introduit en permettant la visibilité des résultats partiels des sous-transactions. Le modèle de transaction par lot s'intéresse à la faiblesse de la bande passante et à la latence des communications sans fil dans un environnement mobile. Le modèle de clusterisation est conçu pour un système totalement distribué pour assurer la consistance de la base de donnée. Le modèle sémantique prend en considération le fait qu'une transaction mobile a une longue durée de vie caractérisée par des délais de transmission longs et des déconnexions non prévisibles, et propose d'améliorer la gestion des accès concurrents tout en préservant la propriété de sérialisation (isolation). Le modèle MDTPM se propose de prendre en charge la continuité de l'exécution des transactions initialisées par les UMs d'une manière transparente durant les déconnexions volontaires de ces dernières. Le modèle Kangourou a été conçu avec le souci de modéliser le mouvement des transactions (ou plutôt leur mobilité d'une cellule à l'autre).

Il faut remarquer que mis à part le modèle de transactions rapporteuses et de co-transactions, tous les autres modèles partent de l'hypothèse que la transaction mobile est initialisée (lancée) par l'UM. A priori, une transaction mobile devrait pouvoir être lancée à partir d'un site quelconque qu'il soit statique ou mobile. Le modèle Kangourou prévoit une exécution sur la partie fixe du réseau, les modèles

MDTPM, de clusterisation, par batch et sémantique permettent une exécution plutôt partagée entre la partie fixe et l'UM.

A ce stade de la recherche aucun des modèles proposés dans la littérature n'est complet. Chacun tente de répondre à seulement un sous-ensemble des caractéristiques ou propriétés requises.

8. Conclusion

De manière générale, les modèles proposés se sont inspiré du modèle de transaction imbriqué ouvert ou multibase de données. Les propriétés ACID classiques ont été supportées « totalement » (par exemple par le modèle sémantique) ou partiellement par les autres modèles (par exemple la durabilité n'est assurée que par le modèle sémantique et le modèle MDTPM).

A ces propriétés s'ajoutent les nouvelles propriétés propres à l'environnement mobile qui sont la **déconnexions** et la **mobilité** (relogement ou relocation). Afin de répondre aux problèmes des déconnexions fréquentes, l'idée du maintien d'un fonctionnement autonome des sites mobiles durant ces périodes fait l'unanimité. Le « caching » des données en local assortie de la notion de consistance flexible, basée sur la sémantique des objets et des opérations, semble incontournable. Nous pensons également que le maintien de la consistance des données dupliquées sur les sites mobiles, peut tirer profit de la facilité de diffusion offerte par les SBs pour la propagation des mises à jour des données à un grand nombre d'utilisateurs faiblement connectés [BAR 97]. La mobilité ou relogement suggère la migration des données et/ou des contrôles. La migration est une notion déjà bien connue dans l'environnement distribué classique mais qui doit être revue dans le contexte des transactions mobiles.

Quel que soit le modèle conçu, les caractéristiques de l'environnement sans fil et mobile font qu'il faut

essayer de déplacer le coût des calculs et des communications sur la partie statique (fortement connectée) du réseau. De plus, il faut prendre en considération le fait que l'espace disque sur l'unité mobile n'est pas fiable et prévoir des enregistrements de sauvegarde sur les sites (ou serveurs) fixes. Cette dernière spécificité, concernant les contraintes liées à l'UM, nous suggérons de l'appeler **propriété de potabilité**.

Références Bibliographiques

- [BAD 95] Nadjib Badache. La Mobilité dans les Systèmes R'partis. Publication Interne N° 962, octobre 1995, IRISA.
- [BAD 98] Nadjib Badache. Ordre Causal et Tolérance aux Défaillances en Environnement Mobile. Thèse de Doctorat d'Etat en Informatique. Présentée à l'USTHB, Octobre 1998.
- [BAL 91] R. Balter, J.P. Banâtre, S. Krakowiak . Construction des Systèmes D'exploitation Répartis. Collection didactique éditée par INRIA, France, 1991.
- [BAR 97] Daniel Barbara. Certification Reports : Supporting Transactions in wireless systems, 17th Int. Conf. On Distributed Computing Systems. Baltimore, Maryland, USA, May 27-30, 1997.
- [CAC 94] Ramon Caceres, Liviu Iftode. The Effects of Mobility on Reliable Transport Protocols. IEEE 1994.
- [CHR 93] Panos K. Chrysanthis. Transaction Processing in Mobile Computing Environment. Proc. Of the IEEE Workshop on advances in parallel and distributed systems, pages 77-83, Princeton, New Jersey, October 1993.

- [DUN 97] Margaret H. Dunham, Abdelsalam Helal, Santosh Balakrishnan. A Mobile Transaction Model That Captures Both The Data And Movement Behaviour. Accepted by the ACM/Baltzer Journal on Special Topics in Mobile Networks and Applications, 1996.
- [GAR 87] Hector Garcia-Molina, Kenneth Salem. SAGA. Proc. of ACM SIGMOD 1987, Int. Conf. on Management of Data, pp 249-259.
- [GRA 81] J. Gray. The Transaction Concept : Virtues and Limitations. In proc. Of the 7th VLDB, Cannes, pp. 144-154, 1981.
- [IMI 92] Tomasz Imielinski and B.R. Badrinath. Querying in Highly Mobile Distributed Environments. Proc. of The 18th VLDB conf. Vancour, British Columbia, Canada 1992.
- [IMI 93] Tomasz Imielinski and B.R. Badrinath. Data Management for Mobile Computing. SIGMOD RECORD, vol. 22, N°1, March 1993.
- [KUM 98] Vijay Kumar, Margaret H. Dunham. Defining Location Data Dependency, Transaction Mobility and Commitment. TR 98-CSE-01, Dept. of Comp. Science and Engineering, Southern Methodist University, Dallas, USA, 1998.
- [LIS 92] James J. Kistler and M. Satyanarayanan. Disconnected Operation in the Coda File System. ACM Transactions on Computer Systems, Vol. 10, N°1, February 1992, Pages 3-25.
- [LU 94] Qi Lu, M. Satyanarayanan. Isolation-Only Transactions for Mobile Operating System Review, pages 81-87, Avril 1994.

- [LU 95] Qi Lu, M. Satyanarayanan. Improving Data Consistency in Mobile Computing Using Isolation-Only Transaction. Proc. of The 5th Work. in Operating Systems, May 4-5 1995, Orcas Island WA.
- [MED 95] B. Medjahed. Transactions imbriquées et contrôle de concurrence. Thèse de Magister en Informatique, USTHB, Alger N° 03/95-M/IN, 1995.
- [MUM 94] L. Mummert, M. Satyanarayanan. Large Granularity Cache Coherence for Intermittent Connectivity. Proc. of the 1994 Summer Usenix Conf., June 8-10 1994, Boston MA.
- [NAR 94] Vivek R. Narasayya. Distributed Transactions in a Mobile Computing System. Tech. Report, Depart. Of Comp. Science, Univ. of Washingto, 8p, March 1994.
- [PIT 94a] Evaggelia Pitoura, Bharat Bhargava. Revising Transaction Concepts for Mobile Computing. Proc. of the IEEE Workshop on Mobile Systems and Applications, Santa Cruz, Ca, Dec 94.
- [PIT 94b] Evaggelia Pitoura, Bharat Bhargava. Building Information Systems for Mobile Environments. Proceedings of the 3rd International Conference on Information and Knowledge Management, Novembre 1994, pp 317-378.
- [PIT 94c] Evaggelia Pitoura, Bharat Bhargava. Maintaining Consistency of Data in Mobile Distributed Environments. Technical Report TR694-25. Department of Computer Science, Purdue University, 1994.
- [PIT 94d] Evaggelia Pitoura and Bharat Bhargava. Dealing with Mobility : Issues and Research Challenges. Technical Report CSD-TR-070,

Depart. Of Computer Science, Purdue
University, November 1993.

- [SCH 95] Werner Schaad, Hans-J; Schek, G. Weikum. Implementation and Performance Of Multi-level Transaction Management in a Multibase Environment. 0-8186-7056-8/95, pp 108-115, IEEE 1995.
- [WAL 84] Bernd Walter. Nested Transactions with Multiple Commit Points : An Approach to the Structuring of Advanced database Applications. Proc. of the Tenth Int. Conference on Very Large Data Bases, Singapore, August 1984.
- [WEI 92] Gerhard Weikum and Hans-J. Schek. Concepts and Applications of Multilevel Transactions and Open Nested Transactions. Chapitre 13, Special supplement to Database Transaction Models For Advanced Applications By Ahmed K. Elmagarmid. Morgan Kaufman Publishers, Inc., 1992.
- [WAL 95] Gary D. Walborn and Panos K. Chrysanthis. Supporting Semantics-Based Transaction Processing in Mobile Database Applications. Proc. Of The 14th Symp. Reliable Distributed Systems, pp : 31-40, Germany, Sept. 1997.
- [WAL 97] Gary D. Walborn and Panos K. Chrysanthis. Pro-motion : Management of Mobile Transactions. Proc. Of The 11th ACM Annual Symposium on Applied Computing, Special Track on Database Technology, pp. 101-108, Van Jose, CA, March 1997.
- [YEO 94] L. H. Yeo, A. Zaslavsky. Submission of Transactions from Mobile Workstations in a Cooperative Multidatabase Processing Environment. The 14th Int. Conf.

Computing Systems Pozuan, Poland June 21-24,
IEEE 1994.

