

3

Trabalhos Correlatos

Este capítulo tem por objetivo apresentar os principais modelos de transações propostos para o ambiente de computação móvel e a motivação para a proposta de um novo modelo de transações.

Inicialmente, serão apresentadas características específicas das transações para o ambiente da computação móvel e, em seguida, diversos modelos de transações de banco de dados propostos para esse ambiente. Finalmente, é enunciada a motivação para a proposta de um novo modelo de transações.

3.1

Transações no Ambiente de Computação Móvel

Em um ambiente de computação móvel, o processamento das *transações* de bancos de dados é afetado pelas constantes *desconexões* dos clientes móveis de suas redes fixas, pela *fraca conectividade* das redes sem fio, pela *mobilidade dos clientes* por um amplo espaço geográfico e pelas constantes *falhas de recuperação* das transações.

A computação móvel precisa ser estruturada como um conjunto de transações, algumas que executam no equipamento móvel, enquanto outras executam no servidor de banco de dados da rede fixa [14].

Além da necessidade do particionamento de suas transações, a computação móvel também está envolvida com a mobilidade de suas fontes de dados e de seus usuários. Assim, uma outra exigência que as transações atômicas não podem satisfazer é a sua habilidade de lidar com falhas parciais e prover diferentes estratégias de recuperação, minimizando dessa forma os efeitos das falhas [14].

Uma **transação móvel** é uma transação distribuída, em que parte da computação é executada no equipamento móvel e parte em um equipamento fixo, não-móvel. O uso do meio sem fio e a mobilidade dos usuários e dos

gerenciadores de dados afetam o processamento das transações de várias formas [46].

O emprego de conexões sem fio pode resultar em transações longas, em função dos longos atrasos (*delay*) da rede. Além disso, *transações móveis* envolvem sites que são conectados intermitentemente ao resto da rede. Por essas razões, os usuários de equipamentos móveis evitam o uso freqüente dos meios sem fio, uma vez que essas conexões são caras, tanto financeiramente quanto em termos do consumo dos recursos dos equipamentos móveis, ou porque os recursos da rede não estão disponíveis na sua localização geográfica.

O custo efetivo da gerência de transação para esse ambiente pode levar à adoção de uma abordagem de suporte a poucas transações longas, ao invés de muitas transações curtas. Finalmente, a computação móvel é mais propensa a erros, em função das freqüentes desconexões e por serem mais suscetíveis a acidentes do que os equipamentos fixos.

A mobilidade resulta em transações que acessam sistemas heterogêneos de informações. Essa heterogeneidade deve-se, principalmente, aos diferentes recursos utilizados nos sistemas, como, por exemplo, capacidade de transmissão das redes, conexões intermitentes, etc. Além disso, em projetos estáticos, a localização dos usuários é fixa, enquanto no ambiente móvel ela muda constantemente. Conseqüentemente, as transações móveis precisam acessar dados de posição (localização) dos clientes, que mudam rapidamente, e, em muitos casos, as posições são imprecisas [46].

3.2

Trabalhos Correlatos

Diversos modelos de transações têm sido propostos para o ambiente de computação móvel. Dentre esses modelos destacam-se: o modelo *Isolation-Only Transactions*, proposto em [37]; o modelo *Two-Tier Replication*, proposto em [26]; o modelo *Mobile Open Nested Transactions*, proposto em [14]; o modelo *Kangaroo*, proposto em [19]; o modelo *Clustering*, proposto em [48, 50]; o modelo *Pro-motion*, proposto em [69]; o modelo *Transaction Processing Agent - TXAgent*, proposto em [10]; o modelo desenvolvido para o projeto *Mobisnap*, proposto em [55]; e o modelo desenvolvido para o projeto *MDRTDBS*, proposto por [34].

A seguir, será apresentada uma breve descrição dos modelos mais significativos, considerando-se como cada um deles trata as propriedades ACID, estendendo o trabalho apresentado em [61] com um maior detalhamento.

3.2.1

O Modelo de Transações Clustering

O modelo de transações *Clustering*, proposto em [48] e estendido em [50], considera um sistema totalmente distribuído, e foi projetado para manter a consistência do banco de dados. O banco de dados é dividido dinamicamente em grupos (*clusters*), cada um agrupando dados semanticamente relacionados ou localizados proximamente. A consistência dos dados é mantida em todos os grupos armazenados nos equipamentos fixos, enquanto diferentes graus de consistência são definidos para os dados, que estão replicados em diferentes grupos armazenados nos equipamentos móveis. Esse grau de consistência pode variar, por exemplo, em função da largura de banda da rede entre os grupos. Os grupos são configurados dinamicamente. Um grupo pode ser distribuído por diversos equipamentos fortemente conectados. Quando um equipamento móvel é desconectado, transforma-se automaticamente em um grupo.

Para cada objeto, duas cópias são mantidas: uma delas (versão rigorosa - *strict version*) tem que ser consistente globalmente e a outra (versão fraca - *weak version*) pode ter algum grau de inconsistência, porém tem que ser consistente localmente. Uma transação móvel, nesse modelo, é então definida como do tipo rigorosa (*strict*) ou fraca (*weak*). Transações fracas acessam somente versões locais de dados e são executadas quando os equipamentos móveis estão desconectados, enquanto as transações rigorosas acessam somente versões de dados consistentes globalmente e são executadas quando os equipamentos estão fortemente conectados.

Nesse modelo, dois tipos de operações são incluídos: leituras fracas (WR - *weak read*) e escritas (gravações) fracas (WW - *weak write*). Transações rigorosas são compostas pelas operações padrões de leitura (SR - *strict read*) e gravação (SW - *strict write*), consideradas operações rigorosas, enquanto as transações fracas são compostas por operações consideradas fracas, executadas nas cópias dos dados armazenados em um mesmo grupo. Quando existe a possibilidade de reconexão ou quando a aplicação requer, um processo de sincronização, executado no servidor do banco de dados, faz com que o banco de dados seja consistente globalmente. Esse modelo necessita de um gerente de transações, no equipamento móvel, para prover execução de transações locais, controle de concorrência, gerência de log e recuperação em caso de falhas.

Propriedade de Atomicidade

Quando o equipamento móvel está conectado, é usado um protocolo de commit para garantir a atomicidade global (por exemplo, Two Phase Commit - 2PC), de que participam vários grupos.

Quando o equipamento móvel está no modo desconectado, as transações são validadas em duas etapas. A primeira etapa é realizada no equipamento móvel, com uma confirmação (commit) local, e a segunda etapa, com um commit na estação de base ou no servidor de banco de dados, na rede fixa.

Na segunda etapa do processo de validação, as transações confirmadas localmente executam commit para tornar as atualizações permanentes no servidor de banco de dados, na rede fixa. A confirmação da transação pode implicar o disparo de mecanismos de reconciliação ou a reexecução de transações. As reconciliações são feitas sintaticamente, e as transações fracas são canceladas ou desfeitas caso suas gravações fracas conflitem com transações rigorosas.

Nesse modelo, podem ocorrer cancelamentos em cascata. No entanto, como transações confirmadas localmente modificam apenas dados locais, somente cancelamentos de transações locais são disparados. Esses cancelamentos referem-se apenas a transações fracas, porque resultados locais estão disponíveis exclusivamente para esse tipo de transação.

Propriedade de Consistência

Esse modelo mantém a consistência de dados replicados com duas versões. Ambas estão localizadas no equipamento móvel, e uma delas (fraca) é usada para dar suporte à evolução dos dados, quando o equipamento móvel estiver no modo desconectado. A segunda versão (rigorosa) sempre tem que estar consistente, mesmo que, algumas vezes, contenha versões antigas (no modo desconectado). Consistência, na versão rigorosa, é preservada usando métodos seriados para uma única cópia, como, por exemplo, *quorum consensus*, cópia mestre, etc.

Informações semânticas são usadas para especificar o grau de inconsistência para as versões fracas com base na semântica da aplicação. Esse grau pode ser restringido, limitando o número de confirmações locais, o número de transações que podem operar sobre cópias inconsistentes, o número de cópias que podem divergir, etc. Existe, também, uma *função h* que controla esse grau, projetando operações rigorosas sobre versões fracas

de dados. A consistência total é alcançada combinando (*merging*) diferentes cópias do mesmo dado, localizado em diferentes grupos (reconciliação).

As transações fracas têm desvantagem em relação às transações rigorosas no processo de ressincronização (reconciliação).

Propriedade de Isolamento

Esse modelo concede visibilidade dos resultados confirmados localmente às transações em execução no mesmo equipamento móvel. Para gerenciar o isolamento (visibilidade moderada), esse modelo propõe novas tabelas de solução de conflitos. Ele utiliza bloqueio em duas fases (two phase locking - 2PL) e propõe quatro tipos de bloqueios que correspondem às operações fracas e rigorosas (WR, WW, SR, SW). Quatro tabelas de conflitos para compatibilidade de bloqueios são propostas. A *função de projeção h* utiliza tabelas de conflitos, para refletir operações rigorosas sobre as versões fracas, dependendo dos requisitos de consistência da aplicação. Por exemplo, consistência rigorosa requer traduzir uma gravação rigorosa (SW) sobre um objeto, em gravações rigorosas (SW) sobre todas as suas cópias (rigorosas e fracas). Conseqüentemente, um bloqueio SW não é compatível com qualquer outro bloqueio. Transações fracas liberam seus bloqueios no commit local e transações rigorosas, no commit global.

Propriedade de Durabilidade

Esse modelo não garante a durabilidade dos efeitos da transação antes da sua confirmação (commit).

3.2.2

O Modelo de Transações Two-Tier Replication

O modelo de transações *Two-Tier Replication*, proposto em [26], considera as abordagens de replicação de dados e de transações para o ambiente da computação móvel, em que os equipamentos móveis estão ocasionalmente conectados. Para cada dado, existem uma versão principal (master) e várias versões replicadas (cópias). A versão master ou controladora armazena o valor atualizado mais recente de um objeto ou item de dado. As réplicas dos dados, armazenadas nos equipamentos móveis, são consideradas provisórias, e são compostas pelos objetos ou itens de dados locais atualizados. O valor

atualizado mais recente de um objeto ou item de dado é mantido como um valor temporário.

Esse modelo suporta dois tipos de transações: as transações base (*base transactions*) e as transações provisórias (*tentative transactions*), que se diferenciam em função dos dados que acessam. Transações base trabalham somente com os dados principais (master) e produzem novos dados desse tipo. Envolvem em suas operações, na maioria dos casos, um nó móvel (desconectado na maior parte do tempo) e diversos nós de base (sempre conectados). Já as transações provisórias trabalham somente com dados temporários locais (provisórios) e produzem novos dados desse tipo. As transações provisórias produzem transações base que serão executadas posteriormente nos nós de base (equipamento fixo).

Em resumo, as transações base são executadas acessando versões master (esquema de replicação postergado do master), enquanto as transações provisórias são executadas acessando versões provisórias (cópias locais). Transações provisórias podem realizar atualizações nos equipamentos móveis, quando esses estão em modo desconectado. Quando a conexão é estabelecida, as transações provisórias são reexecutadas como transações base nos equipamentos da rede fixa (nós de base), para atingir a consistência global do banco de dados. Essa reexecução é a forma de tornar as atualizações locais persistentes no banco de dados da rede fixa.

Esse modelo, assim como o modelo Clustering, necessita de um gerente de transações no equipamento móvel, para prover execução de transações locais, controle de concorrência, gerência de log e recuperação em caso de falhas.

Propriedade de Atomicidade

Esse modelo funciona exatamente como o modelo Clustering, no que se refere à propriedade de Atomicidade. Se transações base falharem (quando estiverem reexecutando transações provisórias), mesmo considerando os critérios da aceitação (anexados em cada transação provisória), as transações provisórias serão canceladas.

Nesse modelo, podem ocorrer cancelamentos em cascata, assim como no modelo Clustering. No entanto, como transações confirmadas localmente modificam apenas dados locais, somente cancelamentos de transações locais são disparados. Esses cancelamentos referem-se apenas a transações provisórias, porque resultados locais estão disponíveis exclusivamente para esse tipo de transação.

Propriedade de Consistência

Esse modelo, assim como o modelo Clustering, mantém a consistência de dados replicados com duas versões. Ambas estão localizadas no equipamento móvel, e uma delas (provisória) é usada para dar suporte à evolução dos dados, quando o equipamento móvel estiver no modo desconectado. A segunda versão (master) sempre tem que estar consistente mesmo que, algumas vezes, contenha versões antigas (no modo desconectado). Consistência, na versão master, é preservada usando métodos seriados para uma única cópia, como, por exemplo, *quorum consensus*, cópia mestre, etc.

Versões provisórias de dados são descartadas na reconexão, desde que elas estejam completamente atualizadas a partir das versões master. As transações provisórias têm desvantagem em relação às transações base no processo de ressincronização (reexecução).

Além das características anteriores, informações semânticas são usadas para estabelecer um critério de aceitação entre as transações base e provisórias.

Propriedade de Isolamento

Esse modelo, assim como o modelo Clustering, concede visibilidade, dos resultados confirmados localmente, às transações em execução no mesmo equipamento móvel.

Propriedade de Durabilidade

Esse modelo, assim como o modelo Clustering, não garante a durabilidade dos efeitos da transação antes da sua confirmação (commit).

3.2.3

O Modelo de Transações Pro-motion

O modelo de transações *Pro-Active Management of Mobile Transactions (Pro-motion)*, proposto em [69] e estendido em [70], é um modelo de transações que atende ao modo desconectado dos equipamentos móveis, atuando em um ambiente cliente/servidor móvel. Esse modelo utiliza um tipo especial de objeto chamado compacto (*compact*), que é a unidade básica de cache e controle nos equipamentos móveis.

Um compacto é um objeto especial que contém, além dos dados para serem armazenados na cache do equipamento móvel, uma extensão que

inclui *obrigações* (como, por exemplo, prazo final - *deadline*), *restrições* (como, por exemplo, um conjunto de operações) e *informações de estado* (como, por exemplo, número de acessos a um objeto). Representa um acordo entre servidor de banco de dados, do equipamento da rede fixa, e o equipamento móvel. Nesse acordo, o servidor de banco de dados delega ao equipamento móvel controle sobre os dados. Já o equipamento móvel concorda em assumir responsabilidades sobre os dados e honrar condições específicas impostas pelo servidor de banco de dados. Informações necessárias para gerenciar o compacto estão encapsuladas nele.

O modelo de transações Pro-motion usa transações divididas aninhadas (*nested-split*) como infra-estrutura. Considera o sistema móvel inteiro como uma transação de longa duração extremamente grande, executada no servidor. Os recursos necessários para criar compactos são obtidos, por essa transação, através de operações usuais de banco de dados. A construção dos compactos é de responsabilidade do *gerente de compacto* (compact manager), no servidor de banco de dados. O gerenciamento dos compactos é realizado por um gerente de compacto, pelo *agente de compacto* (compact agent), no equipamento móvel, e pelo *gerente de mobilidade* (mobility manager), na estação de base. O gerente de compacto agirá como um front-end para o servidor de banco de dados, parecendo ser um simples cliente do banco de dados, executando uma única grande transação, de longa duração.

No entanto, em cada equipamento móvel, o *agente de compacto* realiza o gerenciamento da memória cache, bem como o processamento da transação, controle de concorrência, de log e recuperação. O *gerente de mobilidade* é o responsável pela transmissão entre os agentes. As transações do equipamento móvel são executadas localmente, mesmo quando ele está no modo conectado. O processo de sincronização é executado pelo agente de compacto e pelo gerente de compacto, na reconexão. Esse processo verifica os compactos modificados pelas transações confirmadas localmente. Caso o compacto preserve a consistência global, uma confirmação global é executada.

Propriedade de Atomicidade

Diferentemente dos modelos Clustering e Two-Tier Replication, esse modelo não diferencia se o equipamento móvel está no modo conectado ou desconectado.

As transações, nesse modelo, são validadas em duas etapas. A primeira etapa é realizada no equipamento móvel (commit local) e a segunda etapa

(commit global), na estação de base ou no servidor de banco de dados, na rede fixa.

Na primeira etapa, o commit local é realizado usando um protocolo de commit para um único local. Na segunda etapa do processo de validação, as transações confirmadas localmente executam commit, para tornar as atualizações permanentes no servidor de banco de dados, na rede fixa. Os compactos envolvidos em transações confirmadas localmente são verificados. Se alguns compactos não são mais válidos, as transações são canceladas e um procedimento de contingência (anexado a cada commit local) é executado para obter atomicidade semântica.

Conceitualmente, esse modelo considera as transações como sendo de longa duração. Se essas transações são executadas em sistemas de múltiplos bancos de dados (*multidatabase*), a atomicidade global depende da autonomia de cada sistema de banco de dados. Se alguns SGBDs não podem participar em um protocolo de commit global, então a atomicidade não pode ser garantida.

Nesse modelo, podem ocorrer cancelamentos em cascata, assim como nos modelos Clustering e Two-Tier Replication. No entanto, como transações confirmadas localmente modificam apenas dados locais, somente cancelamentos de transações locais são disparados.

Propriedade de Consistência

Esse modelo explora informações semânticas para construir os compactos. Os compactos representam um acordo entre o servidor de banco de dados, na rede fixa, e os equipamentos móveis. O gerente de compacto e o servidor de banco de dados encapsulam nos compactos: dados, métodos de tipos específicos, informações de estado, regras de consistência e obrigações. Se o agente do compacto e o gerente de compacto respeitam todas essas condições, o uso de compactos não irá afetar a consistência do banco de dados. O projetista do compacto pode determinar critérios de exatidão e métodos de controle de concorrência por compacto.

Propriedade de Isolamento

Esse modelo, assim como os modelos Clustering e Two-Tier Replication, concede visibilidade dos resultados confirmados localmente às transações em execução no mesmo equipamento móvel.

Considerando que esse modelo usa transações aninhadas abertas, o isolamento global não é respeitado, uma vez que as subtransações não são executadas isoladamente. Após o processo de sincronização, as transações de longa duração são divididas. Todas as operações que forem sincronizadas com sucesso formam uma transação separada, que é confirmada no servidor de banco de dados. Resultados dessa transação dividida (confirmada) serão visíveis por todo o ambiente do banco de dados.

Visto que o projetista do compacto pode determinar critérios de exatidão e métodos de controle de concorrência, por compacto, o modelo Pro-motion propõe o uso de uma escala de 10 (dez) níveis de isolamento. Os níveis são caracterizados com base nos graus de isolamento definidos no padrão da SQL ANSI. Quanto maior o nível, maior o grau de isolamento. O nível 9 representa a execução serial de transações, enquanto o nível 0 (zero) não garante qualquer isolamento. Tendo em vista que o uso arbitrário do nível de isolamento pode levar a inconsistência nos dados, foi proposta a seguinte regra:

1. As transações obrigam a um nível mínimo de isolamento para as operações de leitura e gravação;
2. Cada operação está associada a um nível de isolamento;
3. Nenhuma operação de gravação pode ter um nível de isolamento menor do que o nível de isolamento de gravação da transação;
4. Nenhuma operação de leitura pode ter um nível de isolamento menor do que o nível de isolamento de leitura da transação;
5. O menor nível de isolamento de qualquer leitura é maior ou igual ao maior nível de isolamento exigido em qualquer operação de gravação.

Propriedade de Durabilidade

Esse modelo, assim como os modelos Clustering e Two-Tier Replication, não garante a durabilidade dos efeitos da transação antes da sua confirmação (commit). Entretanto, Pro-motion pode dar algumas garantias de durabilidade, embora possam existir condições que não podem ser respeitadas em função das desconexões como, por exemplo, pode existir um prazo no compacto (*deadline*) que não é possível de ser alcançado. Conseqüentemente, durabilidade é difícil de ser obtida no processo de sincronização.

3.2.4

O Modelo de Transações Reporting

O modelo de transações *Reporting*, proposto em [14], analisa transações aninhadas e transações aninhadas abertas (tais como transações Sagas¹, transações divididas e multitransações), apresentando suas limitações para o ambiente da computação móvel.

Esse modelo considera o ambiente de banco de dados móvel como um sistema especial de múltiplos bancos de dados (multidatabase), com necessidades específicas, em que as transações que executam nos equipamentos móveis são consideradas um conjunto de subtransações. Esse modelo considera que as limitações nos equipamentos móveis tornam necessário o uso dos equipamentos fixos, por exemplo, para armazenar parte do estado da computação ou executar parte da computação.

É proposto um modelo de transações aninhadas abertas que suporta *transações atômicas*, *transações não-compensáveis* e dois tipos adicionais de transações: *co-transaction* e *reporting*. *Transações atômicas* têm propriedades padrão de abort e commit; *transações não-compensáveis* delegam, no momento do commit, todas as operações que elas invocaram, para seus pais; *transações reporting* relatam, para outra transação, alguns de seus resultados, em algum ponto durante sua execução. Um relato pode ser considerado uma delegação de estado entre transações. *Co-transactions* são transações do tipo *reporting*, em que o controle é passado da transação reporting para uma outra, que recebe o relato. Transações do tipo *co-transaction* são suspensas no momento da delegação e reiniciam sua execução quando recebem um relato.

Enquanto em execução, transações podem compartilhar seus resultados parciais e manter, parcialmente, o estado de uma subtransação móvel, na estação de suporte à mobilidade, localizada na rede fixa. Essa subtransação deve ter sido executada no equipamento móvel.

Propriedade de Atomicidade

Nesse modelo, cada subtransação é atômica, mas isso não garante a atomicidade da transação móvel global. Para as transações compensáveis, é garantida atomicidade (semântica), uma vez que elas podem ser associadas a subtransações (compensatórias).

¹Transação de longa duração, composta por um conjunto de subtransações relativamente independentes.

Transações não-compensáveis delegam, no momento do commit, todas as operações que elas invocaram, para suas transações pai. Nesse caso, a delegação de transações reporting e co-transaction não afeta a atomicidade, porque não é exigido que a transação que invoca uma operação seja a transação que confirma (commit) ou cancela (abort) a operação.

Conceitualmente, assim como o modelo Pro-motion, esse modelo considera as transações como de longa duração. Se essas transações são executadas em sistemas de múltiplos bancos de dados (*multidatabase*), a atomicidade global depende da autonomia de cada sistema de banco de dados. Se alguns SGBDs não podem participar em um protocolo de commit global, então a atomicidade não pode ser garantida.

Propriedade de Consistência

Nesse modelo, não são propostas novas formas para garantir a consistência. No entanto, subtransações podem ser relacionadas a transações compensatórias, a fim de manter consistência semântica, no caso de cancelamentos. No caso de transações não-compensáveis, esse modelo realiza delegação com base em requisitos semânticos.

Propriedade de Isolamento

Nesse modelo, a visibilidade é permitida nas transações atômicas, reporting e co-transaction, mas não nas transações não-compensáveis. Uma transação atômica pode confirmar (commit) sua execução mesmo antes da confirmação (commit) da transação pai, e suas modificações no banco de dados tornam-se visíveis para outras transações. O objetivo das transações reporting e co-transaction é, precisamente, permitir a visibilidade de resultados parciais durante a sua execução.

Considerando que esse modelo, assim como o modelo Pro-motion, usa transações aninhadas abertas, o isolamento global não é respeitado, uma vez que as subtransações não são executadas isoladamente.

Propriedade de Durabilidade

Nesse modelo, subtransações são duráveis se a transação pai for confirmada (*commit*).

3.2.5

O Modelo de Transações Semantics-based

O modelo de transações *Semantics-based*, proposto em [68], está focado no uso da informação da semântica do objeto para melhorar a autonomia dos equipamentos móveis quando estes estiverem desconectados. Esse modelo utiliza a fragmentação de objetos como solução para lidar com as operações concorrentes e com limitações da capacidade de armazenamento dos equipamentos móveis. Essa abordagem usa a organização dos objetos e a semântica da aplicação para dividir os dados grandes e complexos em fragmentos menores do mesmo tipo. Cada fragmento pode ser armazenado, independentemente, na memória (*cache*) e manipulado assincronamente.

No modelo de transações Semantics-based, as transações móveis são solicitadas no equipamento móvel e, do ponto de vista do servidor de banco de dados, são consideradas transações de longa duração, devido aos retardos (*delay*) da rede. Nenhuma suposição é apresentada sobre a estrutura da transação. A solicitação de fragmento do equipamento móvel inclui dois parâmetros: *critério de seleção* e *condições de consistência*. O critério de seleção indica o dado a ser carregado na memória cache, no equipamento móvel, e o tamanho do fragmento solicitado. As condições de consistência especificam as restrições para preservar a consistência do dado inteiro. A fragmentação dos dados, executada no servidor, permite um controle de concorrência mais apurado. Cópias principais exclusivas de fragmentos são disponibilizadas para os equipamentos móveis, e transações podem ser inteiramente executadas sobre elas. Um processo de reconciliação dos dados é executado pelo servidor quando ocorre uma reconexão. Esse modelo pode ser usado com diferentes tipos de transações.

Propriedade de Atomicidade

Conceitualmente, assim como os modelos Reporting e Pro-motion, esse modelo considera as transações de longa duração. Como os equipamentos móveis são responsáveis pela confirmação local da transação, é possível dar suporte às transações atômicas e não-atômicas. Se essas transações são executadas em sistemas de múltiplos bancos de dados (*multidatabase*), a atomicidade global depende da autonomia de cada sistema de banco de dados. Se alguns SGBDs não podem participar em um protocolo de commit global, então a atomicidade não pode ser garantida.

Propriedade de Consistência

Esse modelo explora informações semânticas para construir fragmentos. Para preservar a consistência, os objetos têm que suportar operações de divisão (*split*) - para criar fragmentos e operações de combinação (*merge*) - para reconciliar fragmentos. Outra restrição para preservar consistência consiste em prover *condições da consistência* (fornecidas pelas aplicações) sobre o objeto inteiro. Essas condições incluem operações permitidas, restrições de seus valores de entrada e condições sobre o estado do objeto.

Propriedade de Isolamento

Esse modelo, assim como os modelos Clustering, Two-Tier Replication e Pro-motion, concede visibilidade dos resultados confirmados localmente às transações em execução no mesmo equipamento móvel.

Para garantir a serialização, transações locais acessam fragmentos da memória cache, através de protocolos convencionais de controle de concorrência, como, por exemplo, o protocolo de bloqueio em duas fases (2PL).

Propriedade de Durabilidade

Esse modelo, assim como os modelos Clustering, Two-Tier Replication e Pro-motion, não garante a durabilidade dos efeitos da transação antes da sua confirmação (*commit*). O modelo Semantics-based garante durabilidade a partir do *commit* local. Ele reduz a disponibilidade dos fragmentos porque pode retê-los por um período de tempo indefinido.

3.2.6

O Modelo de Transações Prewrite

O modelo de transações *Prewrite*, proposto em [39], tenta aumentar a disponibilidade dos dados, no equipamento móvel, pela introdução da operação de pré-gravação ou gravação antecipada (*prewrite*), sem acréscimo à operação padrão de gravação (*write*). Uma pré-gravação torna o valor do dado visível, após uma pré-confirmação (*precommit*), antes da confirmação (*commit*) da transação móvel. As atualizações permanentes no banco de dados são realizadas posteriormente, pela operação de gravação, no momento do *commit*. Duas variantes dos dados são mantidas: pré-gravação (*prewrite*) e gravação (*write*). A variante pré-gravação reflete o estado futuro dos dados, porém pode ser ligeiramente diferente estruturalmente do

valor correspondente gravado. Por exemplo, em um objeto nota fiscal, a pré-gravação refere-se aos produtos vendidos e a gravação refere-se à nota fiscal completa, incluindo os valores dos totais e dos impostos a serem recolhidos.

No modelo de transações **Prewrite**, a idéia principal é dividir a execução da transação entre o equipamento móvel e o servidor de banco de dados. O gerente de transações, no equipamento móvel, executa a transação, porém as atualizações permanentes são feitas no servidor de banco de dados, por um gerente de dados. Esse modelo garante que, por meio da delegação da responsabilidade de gravação no banco de dados, o processamento de transações seja reduzido no equipamento móvel. O modelo Prewrite propõe três operações a serem executadas pelo gerente de transações: operação de *leitura antecipada* (*preread*), operação de gravação ou escrita antecipada (*prewrite*) e operação de confirmação antecipada (*precommit*). Leituras normais e gravações permanentes são realizadas pelo gerente de dados. A estação de base possui capacidade de logging e mantém um relacionamento estreito com o gerente de dados.

A execução da transação é dividida em duas partes. Em um primeiro momento, o gerente de transações solicita à estação de base os bloqueios necessários. A estação de base adquire os bloqueios do gerente de dados. Quando o gerente de transações termina a transação, através de um commit local (*precommit*), gravações antecipadas (*prewrite*) são enviadas para as estações de base. Em um segundo momento, o gerente de dados torna as gravações antecipadas (*prewrite*) permanentes e confirma *commit* a transação móvel. Esse modelo considera as transações móveis de longa duração, e a implementação pode ser realizada por meio de transações divididas e aninhadas.

Propriedade de Atomicidade

Diferentemente dos modelos Clustering e Two-Tier Replication, esse modelo não diferencia se o equipamento móvel está no modo conectado ou desconectado.

As transações nesse modelo, assim como nos modelos Clustering, Two-Tier Replication e Pro-motion, são validadas em duas etapas. A primeira etapa é realizada no equipamento móvel (commit local) e a segunda etapa (commit global), na estação de base ou no servidor de banco de dados, na rede fixa. Através do algoritmo de processamento de transação e do protocolo de bloqueio, o modelo garante que transações confirmadas localmente serão confirmadas, no servidor do banco de dados.

Conceitualmente, esse modelo, assim como os modelos Semantics-Based, Pro-motion e Reporting, considera as transações de longa duração. Se essas transações são executadas em sistemas de múltiplos bancos de dados (*multidatabase*), a atomicidade global depende da autonomia de cada sistema de banco de dados. Se alguns SGBDs não podem participar em um protocolo de commit global, então a atomicidade não pode ser garantida.

Propriedade de Consistência

Esse modelo assegura que o algoritmo de processamento de transação, junto com o protocolo baseado em bloqueio, produz somente escalonamentos seriáveis. Essa serialização é baseada na ordem das confirmações locais das transações móveis. As informações semânticas de objetos são essenciais para garantir consistência nas aplicações móveis. Prewrite define variantes de dados semanticamente idênticos (objetos de pré-gravação/gravação).

Propriedade de Isolamento

Esse modelo, ao realizar o commit local, torna os resultados públicos para todos os equipamentos. Assim como o modelo Clustering, usa protocolo de bloqueio 2PL, e a tabela de operações de conflito inclui operações de leitura antecipada e gravação antecipada (PR - preread, PW - prewrite, R - read e W - write). Como os bloqueios de leituras e gravações antecipadas são gerenciados no nível do gerente de transações e os bloqueios de leitura e gravação, no nível do gerente de dados, não existe conflito entre bloqueios de prewrite/preread e write/read. Para tornar as gravações antecipadas permanentes, o bloqueio de gravação antecipada tem que ser convertido em um bloqueio de gravação, de modo que o gerente de dados possa gravar e confirmar a transação móvel. Bloqueios de leitura antecipada são liberados no momento da confirmação local (commit local), enquanto os bloqueios de gravação antecipada, gravação e leitura o são no momento da confirmação (commit).

Propriedade de Durabilidade

Assim como o modelo Semantics-based, esse modelo garante a durabilidade a partir do commit local. Ele utiliza várias trocas de mensagens para obter bloqueios das estações de base. Se uma transação móvel realiza uma

confirmação (commit) local, é garantida a sua confirmação. Esse modelo não permite que uma transação confirmada localmente seja cancelada.

3.2.7

O Modelo de Transações Kangaroo

O modelo de transações *Kangaroo*, apresentado em [19], propõe um modelo de transações móveis focado no movimento dos equipamentos móveis durante a execução das transações. Esse modelo se baseia em um modelo de transações tradicional, em que as seqüências de operações são executadas sob o controle do SGBD. Nele, as transações móveis são geradas nos equipamentos móveis e são inteiramente executadas em um sistema de múltiplos bancos de dados (MDBS - multidatabase system), situado na rede com fio.

Um agente de software para acesso aos dados foi implementado, estando localizado sobre os gerentes de transação global, já existentes. Esse agente deve ser instalado em todas as estações de suporte à mobilidade para gerenciar as transações móveis e os movimentos dos equipamentos móveis.

No modelo de transações Kangaroo, a preservação das propriedades ACID é de responsabilidade de cada SGBD. O modelo de transações é construído, usando conceitos de transações aninhadas abertas (open-nested) e transações divididas (split). A execução da transação móvel (na verdade, uma transação global) é coordenada pela estação de base na qual o equipamento móvel está conectado no momento. Quando um equipamento móvel se move de uma célula para outra (conseqüentemente mudando de estação de base), a coordenação da transação móvel também se move para a nova estação de base. Essa mobilidade é capturada pela divisão da transação original em duas transações (chamadas de transações *Joey's*, existindo uma transação desse tipo por estação de base). A divisão somente interessa à coordenação da transação. Assim, se o equipamento móvel se move da estação de base 1 para a estação de base 2, a estação de base 1 somente irá coordenar as operações que forem executadas durante a permanência do equipamento móvel na célula da estação de base 1.

Propriedade de Atomicidade

Esse modelo não apresenta novas soluções para as propriedades ACID, uma vez que a execução de suas transações não é afetada pela mobilidade, por ser de responsabilidade do SGBD, localizado no equipamento fixo.

Propriedade de Consistência

Esse modelo não apresenta novas soluções para a propriedade de consistência, no ambiente da computação móvel, usando as definições clássicas para bancos de dados centralizados e distribuídos.

Propriedade de Isolamento

Esse modelo não apresenta características particulares para a propriedade de isolamento de suas transações.

Propriedade de Durabilidade

Esse modelo usa a propriedade de durabilidade convencional para suas transações.

3.2.8

O Modelo de Transações MDSTPM

O modelo de transações *Multidatabase System Transaction Processing Manager* (MDSTPM), proposto em [71], apresenta uma arquitetura para dar suporte a submissões de transação a partir de equipamentos móveis, em um ambiente de múltiplos bancos de dados (multidatabase). Sua contribuição em relação às desconexões dos equipamentos móveis está na implementação do módulo Message and Queuing Facility (MQF), que gerencia a troca de mensagens entre o equipamento móvel e o sistema de múltiplos bancos de dados, na rede com fio. Nesse modelo, é presumido que em cada equipamento móvel e em cada equipamento fixo existe um MDSTPM, no topo do SGBD local. O processamento local é de responsabilidade do SGBD local. O MDSTPM coordena a execução das transações globais, gerando a ordem de prioridade e coordenando as confirmações (commit).

No modelo de transações MDSTPM, assim como no modelo Kangaroo, a maneira como as propriedades ACID são garantidas depende de cada SGBD em cada local. Cada MDSTPM é responsável pela coordenação de suas transações globais. Por causa das desconexões, para cada equipamento móvel tem que ser designado antecipadamente um coordenador na rede fixa. Conseqüentemente, uma vez que um equipamento móvel submeta uma transação global, ele pode desconectar e executar outras tarefas, sem ter que esperar pela confirmação da transação móvel. O equipamento coordenador irá gerenciar a transação móvel em nome do equipamento móvel.

Propriedade de Atomicidade

Esse modelo, assim como o modelo Kangaroo, não apresenta novas soluções para as propriedades ACID, uma vez que a execução de suas transações não é afetada pela mobilidade, por ser de responsabilidade do SGBD, localizado no equipamento fixo.

Propriedade de Consistência

Esse modelo não apresenta novas soluções para a propriedade de consistência, no ambiente da computação móvel, usando as definições clássicas para bancos de dados centralizados e distribuídos.

Propriedade de Isolamento

Esse modelo não apresenta características particulares para a propriedade de isolamento de suas transações.

Propriedade de Durabilidade

Esse modelo usa a propriedade de durabilidade convencional para suas transações.

3.3

A Motivação para um Novo Modelo de Transações

Diversos trabalhos propuseram novos modelos de transações para os bancos de dados utilizados no ambiente de computação móvel. A maioria das propostas se baseia no argumento de que a "manutenção da consistência de todos os dados distribuídos pelos sites impõe um grande *overhead* à computação móvel" [48, 31, 47, 26] e sugere modelos que relaxam algumas propriedades importantes das transações, propriedades essas que garantem a consistência dos dados, diminuindo assim esses *overheads*.

Esses modelos contemplam, basicamente, alguma forma de replicação de dados, mantendo uma versão nos equipamentos móveis e outra nos bancos de dados localizados na rede fixa. Assim, de alguma forma, as consistências local e global dos dados são mantidas, mesmo que em tempos diferentes, por meio de *mecanismos de sincronização* entre os equipamentos móveis e os equipamentos fixos.

Alguns desses modelos são implementados por produtos comerciais, tais como Sybase, Oracle, IBM, DB2 e MS-SQL Server [63, 43, 29, 41]. Uma característica desses modelos é explorar, na maioria dos casos, apenas um tipo de transação (compensável, plana, aninhada, curta, longa, etc.), evitando assim a complexidade dos modelos de transações e diminuindo o *overhead* na computação móvel.

Diferentemente dessa abordagem, que se baseia no *sincronismo* entre os bancos de dados localizados nos equipamentos móveis e nos equipamentos da rede fixa, uma outra abordagem exige que a conexão esteja estabelecida para que ocorra o processamento das transações no SGBD, e o enfoque passa a ser o controle das migrações dos clientes móveis entre as estações de suporte à mobilidade (*handoff*), utilizando transações divididas e compensáveis [19].

No entanto, os modelos convergem quanto ao fato de que não basta apenas um tipo de transação para a definição de *modelos completos de transações* para o ambiente da computação móvel. Pela dinâmica das aplicações desse ambiente, a consistência e atualização dos dados são fatores fundamentais.

O modelo proposto nesta tese trabalha com mais de um tipo de transação, incluindo as transações distribuídas, aninhadas, longas, compensáveis, compensatórias e planas.

Considerando-se os modelos analisados neste capítulo, observa-se que os modelos de transações Clustering e Two-Tier Replication exigem a existência de um SGBD no equipamento móvel, o que nem sempre é desejável, em função dos poucos recursos dos equipamentos móveis.

Por sua vez, os modelos Kangaroo e MDSTPM não podem executar uma transação móvel com o equipamento móvel desconectado, o que é indesejável em função da alta taxa de desconexão dos equipamentos móveis.

A garantia das propriedades ACID, nos modelos Kangaroo e MDSTPM, é efetivada por cada SGBD local, uma vez que a execução das transações não é afetada pela mobilidade dos clientes. Esses modelos usam representantes do cliente móvel, na rede fixa, que cuida da submissão da transação aos SGBDs da rede fixa, via um coordenador global de transações. Nem todo SGBD pode participar desses modelos. Para que um SGBD participe, é necessário que ele suporte 2PC.

Os demais modelos garantem o atendimento das propriedades ACID, de diferentes formas.

Os modelos Pro-motion, Reporting, Semantics-Based e Prewrite geram transações de longa duração, o que aumenta a possibilidade de insucesso

das transações devido a possíveis inconsistências, para os modelos que permitem visibilidade de resultados de transações não confirmadas globalmente (Pro-motion, Reporting e Semantics-Based), ou diminuem a quantidade de transações concorrentes para os modelos que geram bloqueios de recursos (Prewrite). No caso específico do Prewrite, há uma grande quantidade de mensagens trocadas entre o equipamento móvel e as estações de base, o que requer mais conexão.

Os modelos Clustering, Two-Tier Replication e Pro-motion podem gerar cancelamentos em cascata das transações locais, já confirmadas nos equipamentos móveis, em função de conflitos no processo de ressincronização ou reexecução das transações móveis na rede fixa, o que pode afetar o negócio, devido a uma alta taxa de cancelamento de transações.

Enquanto a visibilidade de resultados parciais é interessante para possibilitar maior concorrência de transações, por outro lado aumenta a possibilidade de inconsistências e de transações canceladas em cascata. Os modelos Clustering, Two-Tier Replication, Pro-motion e Semantics-Based permitem a visibilidade de resultados de transações confirmadas localmente para transações em execução no mesmo equipamento móvel. No modelo Reporting, a visibilidade de resultados parciais só é permitida entre as transações especiais reporting e co-transactions. Por sua vez, no modelo Prewrite, a visibilidade dos resultados só é possível após o commit, que gerou bloqueios na rede fixa.

A análise dos modelos apresentados neste capítulo demonstra que nenhum deles consegue atender plenamente a todos os requisitos do ambiente de computação móvel. A proposta deste trabalho consiste em propor um modelo de transações que minimize essas deficiências.

3.4

Resumo do Capítulo

Este capítulo apresentou alguns modelos de transações de banco de dados para um ambiente de computação móvel, bem como uma descrição sucinta sobre esses modelos. Foi enfatizada, a forma como esses modelos implementam e executam suas transações, além de destacado como as propriedades de *Atomicidade*, *Consistência*, *Isolamento* e *Durabilidade* (ACID) são suportadas por esses modelos. As deficiências dos modelos analisados representam a motivação para a proposta de um novo modelo de transações para o ambiente de computação móvel.

No próximo capítulo, serão apresentados a proposta de um modelo de transações para o ambiente de computação móvel, objetivo específico deste trabalho, bem como uma definição formal desse modelo e um diagrama de estados de suas transações durante sua execução.