

UNIVERSIDADE FEDERAL DO CEARÁ
CENTRO DE CIÊNCIAS
DEPARTAMENTO DE COMPUTAÇÃO
MESTRADO EM CIÊNCIA DA COMPUTAÇÃO

Dissertação de Mestrado

**Um Mecanismo de Gerência de Cache para
Servidores Móveis em Redes *Ad Hoc***

Desiere de Sousa Leal

Prof. Javam de Castro Machado

Orientador

Prof^a. Rossana Andrade

Co-orientadora

Fortaleza, Setembro de 2004

UNIVERSIDADE FEDERAL DO CEARÁ
CENTRO DE CIÊNCIAS
DEPARTAMENTO DE COMPUTAÇÃO
MESTRADO EM CIÊNCIA DA COMPUTAÇÃO

Um Mecanismo de Gerência de Cache para Servidores Móveis em Redes *Ad Hoc*

**Este exemplar corresponde à Dissertação de
Desiere de Sousa Leal**

Prof. Javam de Castro Machado
(Orientador)

Prof^a. Rossana Andrade
(Co-orientadora)

Dissertação apresentada ao Mestrado de
Ciência da Computação da Universidade
Federal do Ceará (UFC), como requisito
parcial para a obtenção do título de Mestre em
Ciência da Computação.

Aos meus pais, Rita Laura e Francisco Leal e
ao meu irmão Serley

AGRADECIMENTOS

A Deus, principalmente, que esteve sempre ao meu lado, me dando forças, me protegendo e guiando os meus passos.

À minha família, que sempre me dedicou carinho, cuidado e atenção, me deu forças e aconselhou da melhor forma possível.

Ao professor Javam Machado e à professora Rossana Andrade, pelo papel fundamental de orientação e, ainda, pela atenção e apoio que tiveram comigo durante a construção deste trabalho.

À Fundação Cearense de Apoio à Pesquisa - FUNCAP, pelo incentivo financeiro.

Aos professores e funcionários do Mestrado em Ciência da Computação da Universidade Federal do Ceará, pela contribuição direta ou indireta que deram à realização desse trabalho. À Ranne, funcionária do NPD, pela força e conversas nos horários de espera.

A todos os amigos da turma de Mestrado, pelo aprendizado que tive com discussões e troca de idéias e ainda pelos momentos alegres que ajudaram a descontrair nos momentos de tensão.

Ao longo dos meses de dedicação para a realização deste trabalho, vocês desempenharam papel fundamental para que a conclusão desta dissertação se tornasse possível e, por isso, deixo aqui registrada a minha sincera gratidão.

SUMÁRIO

Agradecimentos	v
Sumário	vi
Lista de Figuras	ix
Lista de Tabelas	x
Lista de Siglas e Abreviaturas	xi
Resumo	xiv
<i>Abstract</i>	xv
1. Introdução	16
1.1. Motivação	16
1.2. Objetivos do Trabalho	16
1.3. Organização do Trabalho	18
2. Ambientes Móveis sem Fio	21
2.1. Características Gerais	21
2.2. Redes Móveis sem Fio	22
2.2.1. Redes de Longa Distância	22
2.2.2. Redes <i>ad hoc</i>	24
2.3. Gerência de Cache em Ambientes Móveis	27
2.4. Conclusão	31
3. Gerenciamento de Dados	32
3.1. Sistemas de Bancos de Dados Móveis	32
3.1.1. Visão Geral	32
3.1.2. Distribuição de Dados.....	33
3.1.3. Gestão de <i>Buffer</i> e Cache	35

3.2. XML	36
3.2.1. Introdução	36
3.2.2. DOM.....	37
3.2.3. Expressões de Caminho (<i>XPath</i>)	39
3.2.4. Dados XML em Ambientes Móveis sem Fio	40
3.3. Conclusão	41
4. Soluções em Gerência de Cache	42
4.1. Gerência de Cache em Sistemas Distribuídos	42
4.2. Gerência de Cache em Ambientes Móveis	43
4.3. Gerência de Cache utilizando XML.....	47
4.4. Discussão	48
4.5. Conclusão	49
5. Gerenciador de Cache para um Servidor de Dados Móvel	50
5.1. Introdução	50
5.2. Arquitetura	51
5.3. O Servidor Móvel	53
5.3.1. Características	53
5.3.2. Armazenamento FoX	53
5.3.3. Serviços de Dados	55
5.4. O Gerenciador de Cache	57
5.4.1. Tratamento de Consultas Dependentes de Localização	57
5.4.2. Interação entre Clientes e Servidor Móvel	61
5.4.3. Estratégias de Substituição	62
5.4.4. Estratégias de Validação	66
5.4.4.1. Tipo de Validação Adotado	66
5.4.4.2. Escopo	67
5.4.4.3. Efeitos em Consultas de Somente Leitura	68
5.5. Manutenção dos Dados em Memória	69
5.6. Conclusão	71
6. Implementação de Protótipo e Experimentos	72
6.1. Configuração do Protótipo do Gerenciador de Cache	73

6.2. Visão Geral do Funcionamento do Gerenciador de Cache	76
6.3. Estratégias e Algoritmos	76
6.3.1. Interação com Clientes Móveis	76
6.3.2. Interação com Servidor Fixo	78
6.4. Estruturas de Manipulação	79
6.4.1. Arquivos	80
6.4.2. Árvores DOM	81
6.5. Funcionamento	81
6.6. Estudos Comparativos	83
6.6.1. Parâmetros de Utilização	84
6.6.2. Resultados Obtidos	87
6.7. Discussão	89
6.8. Conclusão	91
7. Conclusão	92
7.1. Considerações sobre o Mecanismo de Cache Proposto	92
7.2. Trabalhos Futuros	93
7.3. Considerações Finais	93
Referências Bibliográficas	94

LISTA DE FIGURAS

Figura 2.1 - Arquitetura de uma rede celular	23
Figura 2.2 - Mudança de topologia das redes <i>ad hoc</i>	25
Figura 2.3 - Mudança de posição dos dispositivos e da própria rede <i>ad hoc</i>	27
Figura 2.4 - Validação: mudança de localização pode afetar confiabilidade dos dados	29
Figura 2.5 - Substituição: mudança de localização deve ser levada em consideração no cálculo da probabilidade de acesso dos itens em cache	30
Figura 3.1. - Funcionamento da arquitetura de bancos de dados móveis	35
Figura 3.2 - Geração de árvore DOM a partir de documento XML	38
Figura 3.3 - Documento Livraria.xml	38
Figura 3.4. - Representação em DOM de Livraria.xml	39
Figura 4.1 - Arquitetura Sistema Gerenciador de Cache baseado em XML	48
Figura 5.1 - Arquitetura considerada	52
Figura 5.2 - Arquitetura Geral do FoX	54
Figura 5.3 - Gerenciador de Cache no FoX adaptado para Ambientes Móveis	55
Figura 5.4 – Diagrama de Atividade referente à operação de busca de dados pelo servidor móvel de dados	56
Figura 5.5 - Posições dos dispositivos utilizadas no processamento de CDLs	58
Figura 5.6 - Resultado da consulta submetida pelo usuário d1 em posição inicial	59
Figura 5.7 - Resultado da consulta submetida pelo usuário d2, em nova posição da rede <i>ad hoc</i>	60
Figura 5.8 – Movimento da rede e mudança de posição dos dispositivos	60
Figura 5.9 - Gerenciador de Memória do Servidor Móvel	62

Figura 5.10 - Estado do cache após o armazenamento dos itens retornados como resposta à consulta submetida por d1	64
Figura 5.11 - Eliminação em cache para armazenar novos itens	66
Figura 5.12 – Rotina para controle de tempo líquido	69
Figura 5.13 - Manutenção e transporte de dados no formato XML	70
Figura 6.1 - Diagrama de Seqüência - Interação entre entidades	74
Figura 6.2 - Interação entre Clientes Móveis, Gerenciador de Cache e Servidor Fixo	75
Figura 6.3 - Algoritmo de busca em cache	76
Figura 6.4 - Algoritmo de substituição em cache	78
Figura 6.5 - Representação de documentos em Cache	81
Figura 6.6 - Funcionamento do programa simulando interação com clientes móveis	82
Figura 6.7 - Funcionamento do programa simulando interação com servidor fixo	83
Figura 6.8 - Mudança de localização depois de 20 requisições	85
Figura 6.9 - Manutenção de registro dos itens recentemente descartados	86
Figura 6.10 - Gráfico comparativo de percentual de aproveitamento dos dados em cache .	87
Figura 6.11 - Gráfico comparativo da utilização dos itens recentemente descartados	89

LISTA DE TABELAS

Tabela 5.1 - Distância dos itens e quantidade de acesso dos dados em cache	65
Tabela 5.2 – Quadro de desvantagens de cache de páginas	70
Tabela 6.1 – Arquivos utilizados na execução do programa	80

LISTA DE ABREVIATURAS E SIGLAS

ACID	Atomicidade, Consistência, Isolamento e Durabilidade
API	(<i>Application Programming Interface</i>)
BS	(<i>Base Station</i>)
CDL	(Consulta Dependente de Localização)
DDL	(Dado Dependente de Localização)
DM	(Dispositivo Móvel)
DOM	(<i>Document Object Model</i>)
E/S	(Entrada e Saída)
FAR	(<i>Furthest Away Replacement</i>)
GPRS	(<i>General Packet Radio Service</i>)
IEEE	(<i>Institute of Electrical and Eletronic Engineers</i>)
IR	(<i>Invalidation Report</i>)
LAN	(<i>Local Area Network</i>)
LRU	(<i>Least Recently Used</i>)
MRU	(<i>Most Recently Used</i>)
MSC	(<i>Mobile Switching Center</i>)
OID	(<i>Object Identifier</i>)
OODB	(<i>Object Oriented DataBase</i>)
OQL	(<i>Object Query Language</i>)
PC	(Personal Computer)
PDA	(<i>Personal Digital Assistant</i>)
SAX	(<i>Simple API for XML</i>)

SGBD	(Sistema Gerenciador de Banco de Dados)
SGBDD	(Sistema Gerenciador de Banco de Dados Distribuídos)
SQL	(<i>Structured Query Language</i>)
W3C	(<i>World Wide Web Consortium</i>)
XML	(<i>Extensible Markup Language</i>)
XPath	(<i>XML Path Language</i>)
XSLT	(<i>Extensible Stylesheet Language Transformation</i>)

RESUMO

LEAL, Desiere S. **Um Mecanismo de Gerência de Cache para Servidores Móveis em Redes *Ad Hoc***. Orientador: Javam Machado e Co-orientadora: Rossana Andrade, Fortaleza-Ce. UFC, 2004. Diss.

A manutenção de itens de dados freqüentemente acessados na memória cache de um cliente móvel torna-se especialmente importante para melhorar o desempenho do retorno dos dados e a disponibilidade de dados durante períodos de desconexão. Entretanto, a capacidade de locomoção dos usuários em um ambiente móvel introduz uma propriedade conhecida como dependência de localização, que afeta tanto o processamento de consultas quanto as técnicas de gerenciamento de cache.

Dessa forma, um gerenciamento eficiente de cache em sistemas móveis sem fio, deve levar em consideração características adicionais desses sistemas, não presentes em sistemas distribuídos tradicionais, tais como constante mudança de localização do usuário, limitação de recursos em dispositivos portáteis e baixa capacidade das redes sem fio.

Neste trabalho, apresentamos um mecanismo de gerenciamento de cache de dados que atua em um servidor de dados móvel de uma rede *ad hoc*. As estratégias de substituição e validação de cache aqui apresentadas visam agilizar o processamento de consultas dependentes de localização e diminuir o tráfego de dados através da interface aérea de uma rede de longa distância.

Palavras-chave: Gerência de cache, sistemas móveis sem fio, consultas dependentes de localização, redes *ad hoc*, bancos de dados móveis.

ABSTRACT

LEAL, Desiere S. **A Mechanism of Cache Management for Mobile Servers in Ad-hoc Networks.**

Advisor: Javam Machado and Co-advisor: Rossana Andrade, Fortaleza-Ce. UFC, 2004. Diss.

Caching of frequently accessed information in a mobile client's local storage becomes specially important for improving data retrieval performance and data availability during disconnected operation. However, the movement capability of users introduce a property known as location dependency which affect query processing and caching techniques.

Thus, an efficient caching at wireless mobile environment should consider additional characteristics of these systems which are not present in traditional distributed systems, such as constant user's location shifting, limited resources of portable devices and wireless network low capability.

In this work, we present a data caching mechanism which acts in a mobile server of an ad hoc network. The substitution and invalidation strategies aim to reduce location-dependent queries processing time and to decrease data traffic in the wireless interface of a celular network.

Keywords: cache management, mobile systems, location dependent queries, ad hoc networks, mobile databases.

CAPÍTULO 1

INTRODUÇÃO

Neste capítulo introduzimos a motivação, os objetivos e a estrutura geral desta dissertação. Na Seção 1.1 apresentamos a motivação para o desenvolvimento desta dissertação e introduzimos o problema de gerência de cache abordado neste trabalho, na Seção 1.2 são descritos os objetivos do trabalho e, finalmente, na Seção 1.3 é descrita a organização do texto.

1.1. Motivação

A convergência de tecnologias, que envolve o surgimento de computadores portáteis robustos e o desenvolvimento de redes móveis sem fio mais rápidas e mais confiáveis, tem contribuído para a crescente popularidade dos sistemas móveis sem fio. Com isso, verifica-se um vasto interesse em pesquisas nesta área, tanto no meio científico quanto no comercial.

A constante movimentação do usuário, a portabilidade dos dispositivos móveis e o meio de comunicação sem fio fazem com que os sistemas móveis apresentem problemas adicionais, não comuns aos sistemas estáticos tradicionais. Dentre os quais podemos citar:

- Frequentes desconexões, sejam elas por falha da rede, ou para economizar bateria;
- Baixa largura de banda dos canais nas redes sem fio;
- Dispositivos móveis com recursos limitados;
- Consultas que dependem da localização do usuário.

Como um dos requisitos mais importantes no gerenciamento de dados nesse tipo de ambiente é fornecer respostas em tempo-real a transações de aplicações básicas [36], é importante que sejam desenvolvidos métodos computacionais que auxiliem os sistemas móveis, de modo a contribuir para melhorar a taxa de retorno de dados e aumentar a disponibilidade de informações para os usuários desse sistema. Este fato tem contribuído, portanto, para o surgimento de várias pesquisas envolvendo

esses ambientes, em diversas áreas da computação, tais como, sistemas de bancos de dados móveis [3][4][31][65], gerência de transações [15][35], gerência de cache [5][45][55][57][70], segurança [21][46] e outros.

A maioria das áreas citadas anteriormente é afetada pela capacidade de locomoção dos usuários, o que influencia direta ou indiretamente os seus mecanismos de gerência. Como citado em [5], em meios tradicionais de gerenciamento de dados, a relação entre dados e localização geográfica do usuário é normalmente ignorada. Em computação móvel, esta propriedade de “transparência de localização” é, de fato, substituída por uma propriedade conhecida por “dependência de localização”, que será descrita de maneira detalhada no próximo capítulo. Essa propriedade tem influência direta na gerência de dados em ambientes móveis, em particular a gerência de cache, que constitui o foco deste trabalho.

Cache consiste em um conjunto de blocos de dados que estão sendo mantidos em memória por motivos de performance [60]. *Caching* de itens de dados freqüentemente acessados em um cliente móvel torna-se especialmente importante para melhorar o desempenho do retorno dos dados e disponibilidade de dados durante o período de desconexão [45]. Dessa forma, um mecanismo eficiente de gestão de cache melhora consideravelmente o desempenho do sistema, visto que é possível diminuir o tráfego de dados através da interface aérea.

No entanto, as técnicas mais populares de gerenciamento de cache, especificamente as técnicas mais comuns de substituição, tais como LRU (*Least Recently Used*) e MRU (*Most Recently Used*) não são adequadas para sistemas móveis, pois levam em consideração somente a quantidade de acesso dos itens e o tempo decorrido desde o último acesso.

Além do desenvolvimento de estratégias de gerenciamento de cache mais adequadas para o ambiente móvel, nosso trabalho propõe a utilização de itens de dados XML, como modo de manipulação dos dados mantidos em memória cache. Tal proposta foi motivada pela crescente utilização do XML e pela grande quantidade de aplicações para dispositivos móveis aceitando dados nesse formato, possibilitando, desse modo, uma maneira mais estável para troca e manipulação de dados. Essa estabilidade é possibilitada devido ao fato de não ser necessário nenhum tipo de conversão, já que o sistema possui um modelo único de armazenamento e comunicação.

A gerência de cache de dados é particularmente importante em ambientes móveis, pois, além de melhorar a latência de acesso, *caching* de dados pode economizar bateria devido à baixa transmissão de dados e aumentar a disponibilidade de dados em caso de desconexão [43]. Entretanto, o *design* de

mecanismos tradicionais de cache é normalmente baseado na hipótese de estabilidade de rede, e numa razoavelmente alta largura de banda para transmissão. Esses dois requisitos entram em conflito com as características de um ambiente móvel, no qual cada cliente móvel se conecta e desconecta com a rede freqüentemente, e a comunicação entre o cliente móvel e o servidor se faz via um canal de baixa largura de banda e a um custo elevado [45].

Com o intuito de reduzir esses custos, podemos considerar, alternativamente, a utilização das redes móveis sem fio de curta distância, como as redes *ad hoc*. Neste tipo de rede, os dispositivos móveis são capazes de se comunicar entre si por um custo consideravelmente baixo, por utilizarem espectro não licenciado de comunicação e não requererem infra-estrutura fixa. Entretanto, esses aparelhos precisam, freqüentemente, enviar solicitações a um servidor externo através da interface aérea por não possuírem mecanismos eficientes de gestão de memória. Neste sentido, dispondo de um servidor móvel para atuar localmente na rede *ad hoc*, é possível diminuir a quantidade de solicitações ao servidor externo, visto que o servidor móvel fornecerá, aos clientes dessa rede, um mecanismo de gestão de cache adequado para ambientes móveis sem fio.

Em suma, armazenando em cache resultados de consultas prévias, torna-se possível a um dispositivo móvel melhorar o processamento de outras consultas, pois seus resultados, ou parte deles, podem ser obtidos localmente. Verifica-se, porém, que os meios convencionais de gestão de cache não são suficientes para um gerenciamento eficiente dessa área de memória em sistemas móveis, pois não consideram a capacidade de locomoção dos usuários desses sistemas. Desse modo, torna-se necessário considerar a localidade geográfica nas técnicas de gerenciamento de cache implementadas para sistemas móveis sem fio, que constitui a proposta central desta dissertação.

1.2. Objetivos do Trabalho

O principal objetivo deste trabalho é propor técnicas de gerenciamento de cache adequadas para sistemas móveis sem fio, que consideram a localização do dispositivo como um critério relevante para uma eficiente gestão de cache nesses sistemas. Este trabalho foi inicialmente discutido em [42] e, nesta dissertação, temos sua continuação, apresentando-o de forma mais completa.

O gerenciamento proposto tem como ponto central de análise as consultas dependentes de localização e as técnicas apresentadas são implementadas em um servidor de dados móvel que atua numa rede *ad hoc*.

Várias propostas de gerência de cache no cliente móvel têm sido apresentadas, como em [55],[69],[45],[72] e [34], o que melhora consideravelmente o desempenho dos sistemas móveis sem fio. No entanto, dispondo de um servidor móvel, é possível melhorar ainda mais o desempenho nesses sistemas, pois este servidor será capaz de oferecer serviços de gestão de memória, gerência de objetos e pré-processamento de consultas a outros dispositivos móveis, a um baixo custo de comunicação, o que significa uma redução na comunicação entre dispositivos e servidor fixo, ou seja, uma diminuição na taxa de transferência de dados através da interface aérea.

De forma mais específica e resumida, temos como objetivos:

- Oferecer um mecanismo de gerência de cache adequado para ambientes móveis sem fio, considerando a propriedade de dependência de localização, presente nesses sistemas;
- Melhorar o processamento de consultas que dependem da localização do usuário;
- Diminuir o tráfego de dados na interface aérea da rede de longa distância, na qual ocorre a comunicação entre os dispositivos da rede *ad hoc* (dentre eles o servidor móvel) e o servidor externo;
- Aumentar a disponibilidade de dados na rede *ad hoc*, de forma a fornecer suporte em caso de eventuais desconexões com a rede de longa distância;
- Manipular os dados em cache através de itens de dados XML, pois desse modo, não será necessário nenhum mecanismo de conversão de dados, já que as informações já estarão nas estruturas adequadas para serem armazenadas e transmitidas.

1.3. Organização do Trabalho

Este trabalho está organizado em seis capítulos, além desta introdução, descritos a seguir:

No Capítulo 2, **Ambientes Móveis sem Fio**, são delineadas características dos ambientes móveis sem fio, relacionando a elas o contexto de gerência de cache no qual se insere nosso trabalho.

No Capítulo 3, **Gerenciamento de Dados**, são apresentadas noções de gerência de dados em sistemas de bancos de dados móveis e introduzidos conceitos de XML.

No Capítulo 4, **Soluções em Gerência de Cache**, discorre-se sobre vários trabalhos referentes a gerenciamento de cache em ambientes móveis, que forneceram a base fundamental para a elaboração desta dissertação.

No Capítulo 5, **Gerência de Cache para um Servidor de Dados Móvel**, são apresentados os mecanismos e estratégias utilizadas pelo gerenciador de cache proposto, que atua em um servidor de dados móvel.

No Capítulo 6, **Implementação de Protótipo e Experimentos**, é demonstrada a implementação do gerenciador de cache, apresentando suas configurações, principais algoritmos e estudos comparativos, utilizados para validar as estratégias propostas.

No Capítulo 7, **Conclusão**, são apresentadas as conclusões e as contribuições deste trabalho, bem como suas perspectivas futuras.

CAPÍTULO 2

AMBIENTES MÓVEIS SEM FIO

Este trabalho é fundamentado nas características dos ambientes móveis sem fio e nas conseqüências que essas características podem trazer para o gerenciamento de cache nesse tipo de sistema. Dessa forma, neste capítulo são apresentadas algumas características básicas de sistemas móveis sem fio, fornecendo assim, uma base para nossa pesquisa. Além disso, apresentamos o contexto no qual se insere nosso trabalho em relação a esse tipo de ambiente. Na Seção 2.1 são identificadas algumas das características básicas dos ambientes móveis sem fio; na Seção 2.2 são introduzidos os tipos de redes móveis que são abordados no trabalho e, finalmente, na Seção 2.3, é apresentada uma discussão sobre a gerência de cache nesse ambiente.

2.1. Características Gerais

Dois fatores principais contribuem para que sistemas móveis sem fio seja um tema de grande relevância nos dias de hoje [61]. São eles:

- Os crescentes avanços na tecnologia de equipamentos portáteis;
- O interesse intenso em comunicação sem fio por parte de indústrias e usuários.

Contudo, apesar dos avanços tecnológicos e da crescente popularidade desses sistemas, algumas propriedades que lhe são inerentes acrescentam complexidade na gerência desses sistemas, como por exemplo, no suporte a bancos de dados, nos mecanismos de segurança de transmissão de dados, na gerência de cache, entre outros.

As principais dificuldades encontradas nos sistemas móveis sem fio originam-se de três propriedades essenciais da computação móvel: comunicação sem fio, mobilidade e portabilidade [18]. Discutimos nos parágrafos seguintes, a respeito de cada uma dessas propriedades.

As redes móveis sem fio se comunicam através de ondas de rádio ou pulsos de luz infravermelha. A comunicação sem fio é exposta a mais obstáculos que a comunicação fixa, pois os dados

envolvidos interagem com mais sinais e caminhos bloqueados, introduzindo ruídos e ecos. Como resultado, a comunicação sem fio é caracterizada por baixas larguras de banda, altas taxas de erro e freqüentes desconexões.

Em relação à mobilidade, a capacidade de mudar de localização enquanto conectados à rede aumenta a volatilidade de algumas informações. Certos dados considerados estáticos em redes tradicionais, tornam-se dinâmicos em um ambiente móvel, como por exemplo, um computador estacionário pode ser configurado estaticamente para preferir o servidor mais próximo, enquanto um dispositivo móvel necessita de um mecanismo para determinar que servidor ele deve utilizar em um certo momento. A mobilidade também acrescenta obstáculos no processamento de consultas dependentes de localização (que será discutido de maneira mais aprofundada em capítulos subseqüentes) e no gerenciamento de cache, ambos temas centrais da nossa pesquisa.

A propriedade de portabilidade refere-se à característica inerente aos dispositivos móveis de serem equipamentos portáteis, isto é, equipamentos que podem ser transportados de um lugar para outro. Essa propriedade de portabilidade traz consigo problemas referentes à segurança, como riscos de acesso não autorizado ou de alterações de dados, e, ainda, preocupação com consumo de bateria, visto que é importante reduzir pesos dos dispositivos e, conseqüentemente das baterias, o que diminui seu poder computacional. Com os constantes avanços tecnológicos nessa área, espera-se que cada vez mais, venham surgir equipamentos versáteis e poderosos e, por este motivo, não dedicamos atenção especial à característica de portabilidade neste trabalho.

2.2. Redes Móveis sem Fio

2.2.1. Redes de Longa Distância

Redes de longa distância consideradas móveis sem fio são, em geral, redes celulares ou Internet móvel, entretanto, quase todos os sistemas móveis de longa distância são celulares. A Internet móvel possui comportamento semelhante ao das redes celulares, porém, por não estar ligada ao nosso trabalho, não fornecemos nenhuma discussão sobre esse assunto.

Os sistemas celulares dependem de uma rede de células, com poderosos transceptores de rádio no centro de cada célula [14]. A infra-estrutura das redes celulares é complexa e não será discutida de

maneira aprofundada neste trabalho. Entretanto, é importante entender que um ambiente celular é idealizado como uma coleção de áreas geométricas chamadas células, cada uma assistida por uma estação-base (BS - *Base Station*), localizada em seu centro. Uma ou mais células são ligadas a uma MSC (*Mobile Switching Centre*) que atua como um *gateway* da rede celular à rede fixa [12].

Para um melhor entendimento dessa estrutura, na Figura 2.1, são ilustrados os principais componentes da arquitetura de uma rede celular, citados no parágrafo anterior.

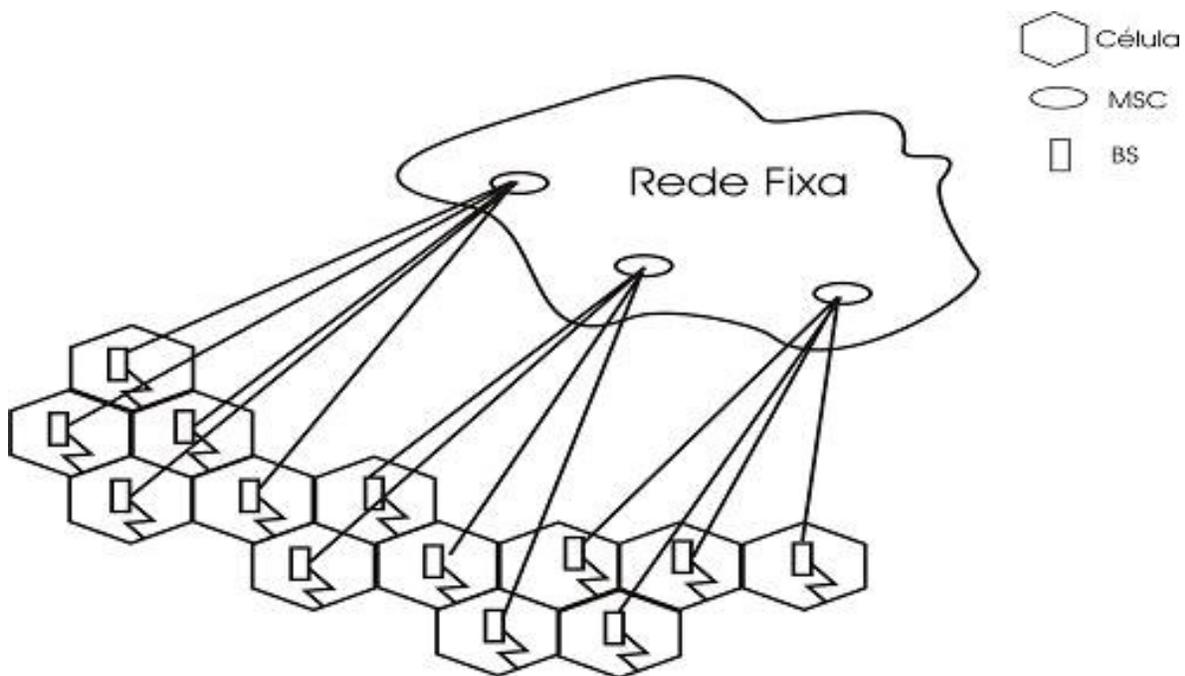


Figura 2.1 – Arquitetura de uma rede celular [12]

Como pode ser verificado nesta figura, a BS é localizada no centro de cada célula e é responsável pelo gerenciamento da interface aérea entre o dispositivo móvel e a estação-base, enquanto a MSC é o elemento de controle da rede móvel sem fio, sendo responsável pelo gerenciamento de chamadas às células, monitoramento de tráfego, gerenciamento geral da rede, entre outros. Essa arquitetura possibilita a comunicação entre dispositivos móveis e de dispositivos móveis com redes fixas, mesmo através de longas distâncias.

A comunicação das redes celulares é feita através da utilização de espectro licenciado de rádio e, apesar dos constantes avanços tecnológicos das redes móveis sem fio, esse espectro continua sendo um recurso limitado e o processo de licenciamento permanece sendo uma questão controversa com governos, empresas e outros interessados, incapazes de concordar sobre como o espectro deve ser dividido [14].

Nesse sentido, os usuários das redes celulares são obrigados a pagar altos preços pela sua utilização, visto que necessitam e operadoras, que adquirem licenças de espectro e disponibilizam os serviços de comunicação necessários.

A crescente demanda por sistemas de comunicação sem fio requer serviços melhores, mais rápidos e mais flexíveis. Esta necessidade de operar e crescer indefinidamente com uma alocação limitada de canais faz do balanceamento de carga e dos esquemas de atribuição de canais, mecanismos importantes para tratar a limitação de frequência de espectro [12].

Além do balanceamento de carga e dos esquemas de atribuição de canais, outra forma de melhorar a utilização de canais é maximizar o reuso de informações pelos dispositivos móveis, a fim de diminuir a comunicação entre estes dispositivos e o servidor da rede fixa. Dessa forma, reutilizando informações entre dispositivos móveis, é possível diminuir o fluxo de dados na interface aérea da rede celular, diminuindo a utilização dos canais de transmissão e, conseqüentemente, reduzindo custos de comunicação.

2.2.2. Redes *Ad Hoc*

As redes *ad hoc* vêm despertando profundo interesse nos últimos tempos, devido, principalmente, ao seu relativamente baixo custo de comunicação, pois essas redes utilizam espectro não-licenciado para transmissão de dados. Tal interesse pode ser visto tanto por parte de executivos, que não precisam ficar presos às mesas de escritório, quanto por usuários comuns, que desejam trocar informações entre vários dispositivos que se encontram próximos. Motivados por essa crescente popularidade das redes *ad hoc*, esse tipo de rede móvel sem fio constitui o foco do nosso trabalho.

Uma rede *ad hoc* consiste em nós móveis com poder de transmissão similar e que não requer infra-estrutura fixa de acesso. Nó móvel é um outro termo utilizado para representar um dispositivo conectado por uma rede móvel, cuja localização pode ser modificada. Qualquer dispositivo com um microprocessador é um nó em potencial para uma rede *ad hoc*.

Diferente das redes sem fio tradicionais e das redes fixas, as redes *ad hoc* não podem confiar em rotas especializadas para descoberta de caminhos e rota de tráfego, pois sua topologia pode ser constantemente modificada, devido à movimentação dos usuários (vide Figura 2.2). Nesta figura, os nós A, B, C, D, E e F constituem uma rede *ad hoc*. O círculo representa a faixa de rádio do nó A. Quando o nó D move-se para fora da faixa de rádio de A, a topologia da rede se modifica.

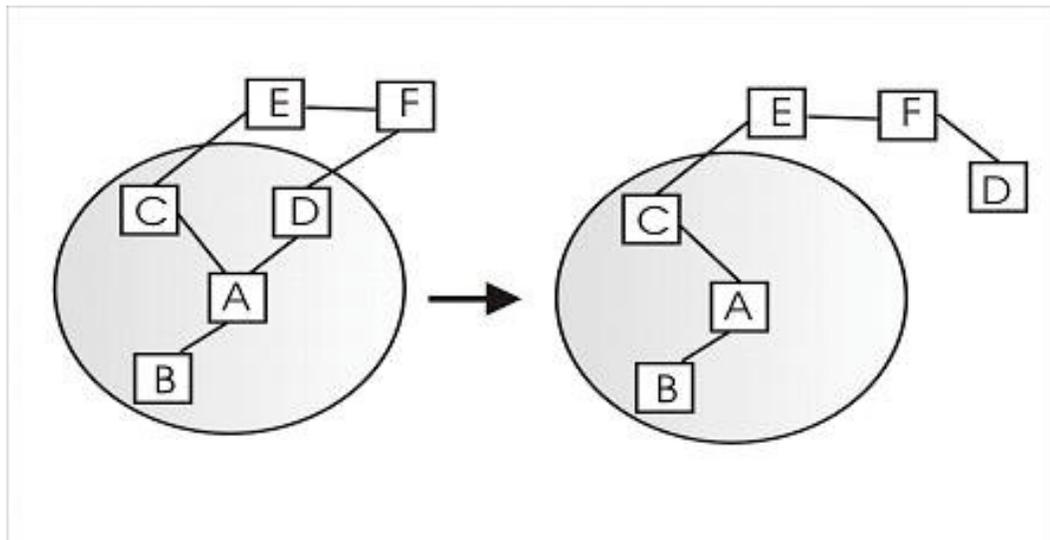


Figura 2.2 - Mudança de topologia das redes *ad hoc*

Verifica-se que, conseqüentemente, sistemas computacionais móveis em redes *ad hoc* devem atuar cooperativamente para criar rotas de tráfego e adaptar a rede para o estado dinâmico de seus *links* e seus padrões de mobilidade [50].

Espera-se que as redes *ad hoc* venham desempenhar, futuramente, importante papel em ambiente comercial e militar. Aplicações potenciais para esse tipo de rede incluem infra-estrutura de rede imediata para suportar computação colaborativa em ambientes móveis, monitoração de pacientes de cuidados críticos, comandos distribuídos e sistemas de controle e acesso móvel à Internet.[23]. Em todos os casos, verificamos a utilidade de um eficiente gerenciamento de cache disponível em um servidor de dados móvel, tema central deste trabalho, no sentido de manter próximos os dados mais prováveis de serem requisitados pelos usuários conectados entre si em uma rede *ad hoc*. Tais usuários

estão geralmente em um contexto similar, ou seja, possuem interesses em comum, o que implica que dados referentes à resposta de um cliente podem ser do interesse de outro.

Para exemplificar esse contexto similar, podemos considerar um grupo de gerenciamento de tráfego, no qual vários usuários da rede *ad hoc* fazem solicitações de informações referentes às condições de tráfego de determinada área. Nesse grupo, é provável que informações possam ser reutilizadas entre os usuários, de maneira que o reaproveitamento de dados disponíveis localmente na rede *ad hoc* intensifica a sua utilização e, fazendo uso da maior quantidade possível desses dados disponíveis, é possível diminuir, conseqüentemente, as requisições através de uma rede de longa distância.

As tecnologias mais utilizadas em redes *ad hoc* são as LANs (*Local Area Networks*) sem fio e as redes *Bluetooth*. As LANs sem fio, padronizadas como IEEE 802.11 podem alcançar velocidades de 125 Mbps, utilizam os mesmos protocolos de comutação da *Ethernet* e possuem abrangência média de 20 m em ambientes internos e 60 m em ambientes externos. O *Bluetooth* é uma tecnologia recente e promete incluir um sistema sem fio completo em um único chip, suficientemente barato para ser embutido em todos os telefones móveis, PCs, PDAs e outros dispositivos [14].

É importante perceber que, apesar da pequena abrangência das redes *ad hoc*, a propriedade de mobilidade dos dispositivos, descrita na Seção 2.1, ainda deve ser considerada. A característica de mudança de localização dos dispositivos portáteis continua sendo crítica, pois além da locomoção dos dispositivos dentro da própria rede *ad hoc*, existe ainda a locomoção da rede como um todo, já que esta rede também tem a característica de ser móvel.

Na Figura 2.3 apresentada a seguir, esse comportamento pode ser observado claramente. O círculo representa a rede *ad hoc* e as setas direcionadas indicam mudança de posição. Através da figura, podemos observar que dois tipos de locomoção são possíveis: a mudança de localização dos dispositivos dentro da rede (como ilustrado pelos nós A, B, C e D) e a mudança de localização da própria rede *ad hoc*.

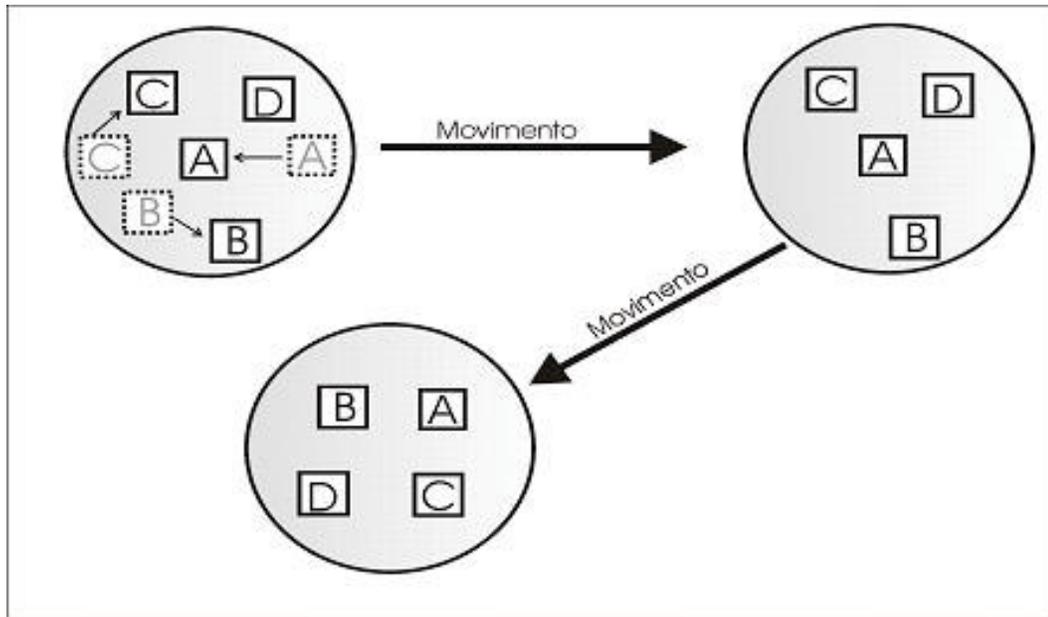


Figura 2.3 - Mudança de posição dos dispositivos e da própria rede *ad hoc*

2.3. Gerência de Cache em Ambientes Móveis

Como discutido brevemente no Capítulo 1, os dados em cache podem ser afetados em virtude da capacidade de locomoção do usuário. Dessa forma, os mecanismos de gerência de cache devem ser adaptados para que se tornem adequados para utilização em ambientes móveis sem fio.

Em ambientes de computação móvel, a propriedade de dependência de localização acarreta problemas adicionais em diversas áreas de gerenciamento de dados em sistemas móveis sem fio, dentre elas a gerência de cache, que constitui o tema central deste trabalho.

Neste sentido, as técnicas tradicionais de gerenciamento de cache não são suficientes para uma hábil utilização do cache, visto que tais estratégias utilizam como critérios de substituição em cache somente a quantidade de acessos e/ou o tempo decorrido desde a última utilização do item e consideram, nos processos de validação, somente a possibilidade de modificação do conteúdo do dado. Desse modo, em sistemas móveis, faz-se necessário considerar características adicionais às já existentes nas técnicas de gerência de cache tradicionais, como por exemplo, localização do dispositivo e/ou direção de movimento.

A gerência de cache é normalmente subdividida em dois subproblemas, o de validação e o de substituição e, em ambos, a mudança de localização de um dispositivo móvel pode afetar o seu funcionamento. Descrevemos detalhadamente, a seguir, cada um desses dois subproblemas, analisando em que pontos a locomoção do usuário deve ser considerada.

- **Validação**

Os métodos de validação em cache devem garantir que os dados disponíveis nessa região da memória sejam confiáveis, ou seja, que os conteúdos desses dados estejam corretos. Manter a consistência do cache é um grande problema, pois, em geral, as informações em cache tornam-se obsoletas se estas forem atualizadas no servidor (no caso de sistemas cliente-servidor) ou modificadas em algum dos diversos *sites* (no caso de sistemas distribuídos). Dessa forma, tanto em sistemas cliente-servidor como em sistemas distribuídos, se um dado for modificado por alguma entidade, o conteúdo desse dado no cache das outras entidades pode ficar comprometido. Felizmente, eficientes métodos de validação têm sido implementados para garantir a confiabilidade dos dados nesses sistemas.

Em sistemas móveis sem fio, adicionalmente, os dados podem tornar-se obsoletos quando o usuário muda de localização. Para tornar mais clara tal situação, suponhamos que um usuário submete uma consulta, cujo resultado depende da sua localização corrente (consultas desse tipo serão detalhadas no Capítulo 5), como, por exemplo, “Retorne os restaurantes mais próximos a um raio de 5 Km”. Após o recebimento da resposta a essa consulta, os dados contidos nesse resultado são, então, armazenados em seu cache local. Se, em seguida, esse usuário muda de posição, o conteúdo preservado em seu cache pode se tornar inútil.

Consideremos a Figura 2.4 como forma de ilustrar tal problema. Na posição inicial, o usuário submete a consulta exemplificada no parágrafo anterior, armazenando em cache o resultado retornado em resposta a essa consulta (no exemplo, restaurantes A e B). O círculo pontilhado representa o raio de proximidade dos restaurantes. Em seguida, o usuário se locomove para uma nova posição, como indicado pela seta e submete novamente a mesma consulta. Verificamos, porém, que os dados mantidos em cache não devem ser considerados como resposta a essa consulta, visto que se tornaram inválidos após a locomoção do dispositivo.

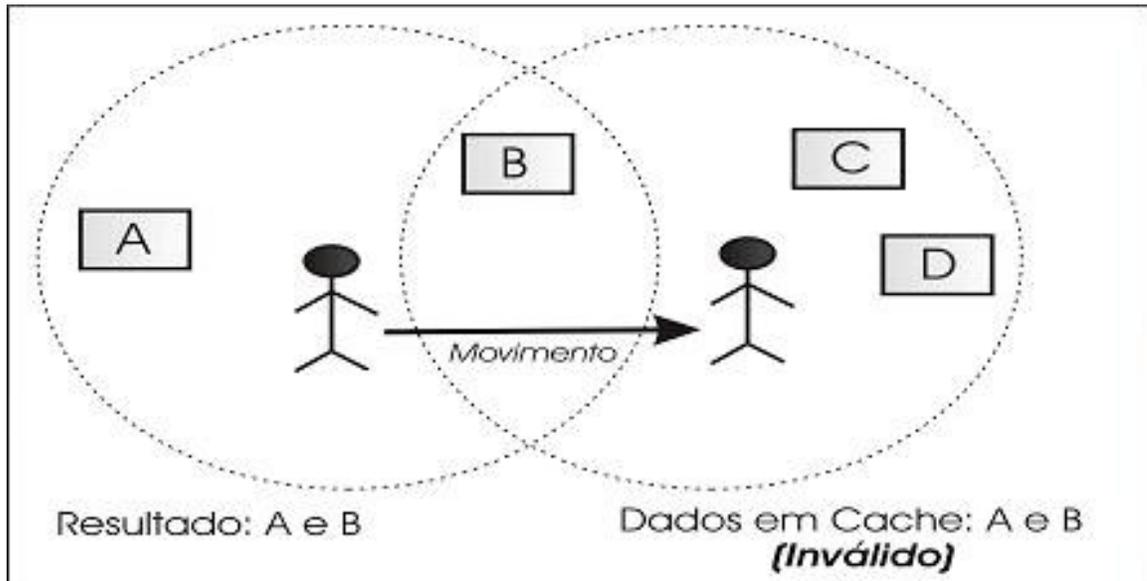


Figura 2.4 - Validação: mudança de localização pode afetar confiabilidade dos dados

Uma maneira de promover validação de cache nas redes móveis é vincular o escopo válido para uma instância de dados retornada ao cliente. O escopo de um item corresponde à área geográfica na qual o valor deste item é considerado válido, de maneira que essa informação é utilizada para garantir a consistência dos dados armazenados em cache.

Dessa forma, o cliente armazena os dados e seu escopo válido para, posteriormente, checar sua validade [43]. Como será mostrado detalhadamente no capítulo 5, utilizamos essa noção de escopo para validar os dados no cache do servidor móvel.

- **Substituição**

As técnicas de substituição em cache devem ser aplicadas quando há necessidade de armazenar novos dados e não há espaço suficiente para fazê-lo. Essas técnicas devem ser implementadas de modo a eliminar do espaço de memória reservada para cache os itens que possuam menor probabilidade de acesso futuro.

As estratégias mais populares em sistemas computacionais tradicionais, tais como LRU e MRU, levam em consideração a quantidade de acessos e a marca de tempo do último acesso de um

determinado item enquanto mantido em cache. Contudo, além dessas estratégias baseadas no tempo decorrido desde a última utilização do item ou no número de acessos desse item, a substituição em cache em um ambiente móvel deve levar em consideração, ainda, a localização do usuário, assim como a localização dos itens armazenados. Tal procedimento faz-se necessário devido à capacidade de locomoção dos usuários, como já discutido anteriormente.

A Figura 2.5, a seguir, ilustra um exemplo no qual a propriedade de localização do usuário deve ser levada em consideração, a fim de se determinar quais dados têm menor probabilidade de acesso futuro.

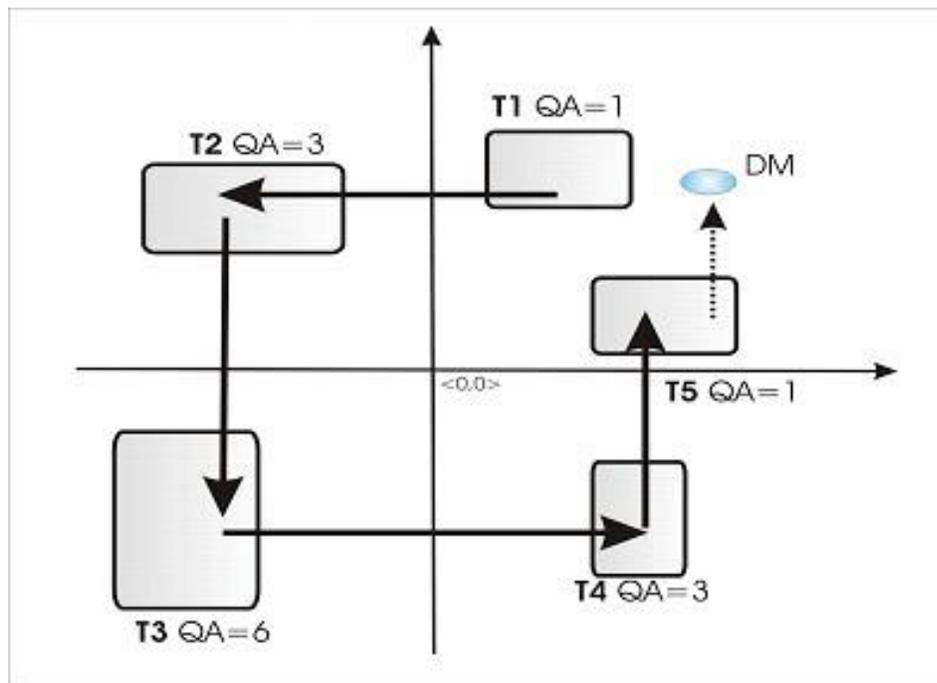


Figura 2.5 - Substituição: mudança de localização deve ser levada em consideração no cálculo da probabilidade de acesso dos itens em cache

Nessa figura, DM indica o dispositivo móvel e os retângulos constituem blocos de dados retornados como respostas a consultas dependentes de localização, os quais foram armazenados em cache no instante T indicado. Considere que as setas representam o trajeto de movimento do dispositivo

móvel, QA indica a quantidade de acessos dos itens em cache e T indica o instante de tempo no momento em que determinada consulta foi submetida, onde $T1 < T2 < T3 < T4 < T5$.

Se no instante T5 o usuário submete uma nova consulta e o espaço em cache não é suficiente para armazenar o resultado, é necessário, portanto, eliminar algum bloco de dados. Utilizando as técnicas tradicionais de substituição, o bloco gerado em T1 seria eliminado, já que é o mais antigo e possui menos acessos. Verificamos, entretanto, que os itens de dados presentes nesse bloco são mais prováveis de serem acessados se considerarmos a propriedade de localização, pois tais itens encontram-se mais próximos da posição corrente do usuário.

No nosso gerenciamento de cache, utilizamos como critério de maior relevância na substituição em cache, a posição dos itens, visto que nosso mecanismo de gerência atua em um servidor de dados móvel de uma rede sem fio e possui, portanto, a propriedade de dependência de localização (vide Capítulo 5).

2.4. Conclusão

Neste capítulo, descrevemos as características essenciais dos sistemas móveis sem fio e introduzimos os problemas de gerência de cache causados pela característica de dependência de localização presente nesses sistemas.

Com o intuito de fornecer um conjunto de informações básicas para um bom entendimento deste trabalho, adicionamos aos conceitos de sistemas móveis sem fio discutidos neste capítulo, definições e características de bancos de dados móveis e itens de dados XML, que serão apresentados no capítulo seguinte.

CAPÍTULO 3

GERENCIAMENTO DE DADOS

Neste capítulo são apresentadas algumas características de sistemas de bancos de dados móveis, com o intuito de facilitar o entendimento do nosso trabalho no que se refere ao tratamento de dados em sistemas móveis. Além disso, é apresentada uma introdução ao XML, visto que as informações no servidor móvel são armazenadas e manipuladas neste formato. Na seção 3.1 é apresentada uma breve discussão sobre sistemas de bancos de dados móveis e na seção 3.2 são introduzidos conceitos básicos de XML.

3.1. Sistemas de Bancos de Dados Móveis

Em um ambiente de computação móvel, a necessidade de um sistema de gerenciamento de banco de dados de tempo-real é crítica, pois um dos requerimentos no gerenciamento de dados nesse ambiente é fornecer respostas em tempo-real a transações de aplicações básicas [36]. Diante deste fato, descrevemos, a seguir, algumas características essenciais que devem ser consideradas no suporte a bancos de dados móveis, como forma de esclarecer pontos importantes desses sistemas e fundamentar alguns conceitos e funcionalidades que devem ser entendidas para um suporte ao trabalho que está sendo apresentado.

3.1.1. Visão Geral

Os sistemas de bancos de dados elaborados sobre redes móveis sem fio são conhecidos por bancos de dados móveis e se caracterizam por atuarem em um ambiente no qual a mobilidade de alguns equipamentos faz com que dados estáticos em redes estacionárias se tornem dinâmicos e voláteis em redes móveis.

As dificuldades acarretadas por essa característica de mobilidade em ambientes móveis (apresentadas no Capítulo 2) despertam diversas questões para projeto de sistemas gerenciadores de bancos de dados distribuídos, pois elas causam mudanças significativas na infra-estrutura sobre a qual tais sistemas são construídos. Sendo assim, muitas das funções dos SGBD's são afetadas em graus variados.

Nas seções seguintes, destacamos, de forma detalhada, as questões de distribuição de dados e gestão de buffer e cache em sistemas de bancos de dados móveis, por serem estas as questões mais relevantes para este trabalho. No entanto, além dessas, várias outras questões de gerenciamento de dados são afetadas nos sistemas móveis sem fio e, portanto, também devem ser consideradas, quando se analisam funções de gerenciamento de dados em ambientes móveis sem fio, tais como:

- Gerenciamento de diretório, no qual surge a dificuldade de como localizar uma estação móvel que possa conter os dados exigidos;
- Difusão de dados, que deve considerar a restrição de que estações móveis podem estar desconectadas no momento de receber os dados difundidos;
- Processamento e otimização de consultas, pois as consultas em ambientes móveis podem ser “dependentes de localização” (essas consultas são detalhadas no Capítulo 5) e a mobilidade de estações de acesso torna difícil a determinação de custos de comunicação entre a estação de acesso e a estação em que os dados residem, dificultando, assim, otimização de consultas;
- Gerenciamento de transações, no qual as técnicas típicas de controle de concorrência, baseadas em bloqueio, podem não ser suficientes, pois redes sem fio são mais propensas a falhas. Além disso, as propriedades ACID podem ser comprometidas devido à mobilidade e às freqüentes desconexões.

3.1.2. Distribuição de Dados

A cada dia cresce a utilização de computadores portáteis e PDAs (*Personal Digital Assistants*) e, conseqüentemente, em um futuro próximo, mais e mais aplicações irão residir em sistemas móveis sem fio. Várias dessas aplicações, tais como bancos de dados, necessitam da habilidade de baixar

informações de um repositório e operar com essas informações até mesmo quando o cliente está desconectado.

Podemos citar como exemplo deste cenário um ambiente onde usuários necessitam acessar e atualizar informações ou de arquivos do seu diretório local, ou de um servidor de banco dados. O tipo de acesso e carga de trabalho gerados por tais usuários é significativamente diferente da carga de trabalho que ocorre nos sistemas cliente-servidor de hoje, devido às características de mobilidade e restrições computacionais presentes nos sistemas móveis sem fio [3].

Podemos considerar, portanto, uma distribuição de dados em ambientes móveis, fato este que adiciona complexidade ao tratamento de informações em tais sistemas. Neste cenário, identificamos uma importante característica na qual alguns clientes acumulam dados, trabalhando neles localmente de modo desconectado. Porém, em um determinado momento tais dados devem ser reintegrados para garantir a confiabilidade do sistema.

Baseado na característica exposta no parágrafo anterior, em [3], é proposto um modelo de arquitetura para bancos de dados móveis, que consiste em uma arquitetura cliente-servidor modificada. Neste modelo, um servidor de banco de dados centralizado atende a requisições dos clientes móveis e alguns desses clientes podem acumular informações em seus discos, criando réplicas locais.

Esses clientes que criam réplicas de dados são conhecidos por *hosts* móveis e o funcionamento de tal arquitetura é ilustrado através da Figura 3.1 a seguir. Como ilustrado na figura, quando os clientes estão conectados, as consultas são processadas pelo servidor e, quando se desconectam, um servidor local no próprio cliente utiliza os dados das réplicas para atender às requisições. Quando o cliente se reconecta, os servidores locais conciliam suas réplicas com a cópia do servidor, reintegrando quaisquer atualizações locais.

Um SGBD móvel é simplesmente um sistema de gerenciamento de dados adaptado para funcionar, adequadamente, em um ambiente móvel sem fio. De modo semelhante aos SGBDs distribuídos convencionais, um SGBD móvel deve ser capaz de se recuperar de falhas de comunicação e, apesar de ser mais exposto a falhas do que os ambientes computacionais não-móveis (devido às restrições de capacidade de bateria e redes de comunicação), algumas das falhas podem ser previsíveis, o que capacita o sistema a agir eficientemente contra tais falhas, garantido, desse modo, a confiabilidade dos dados nesses sistemas [36].

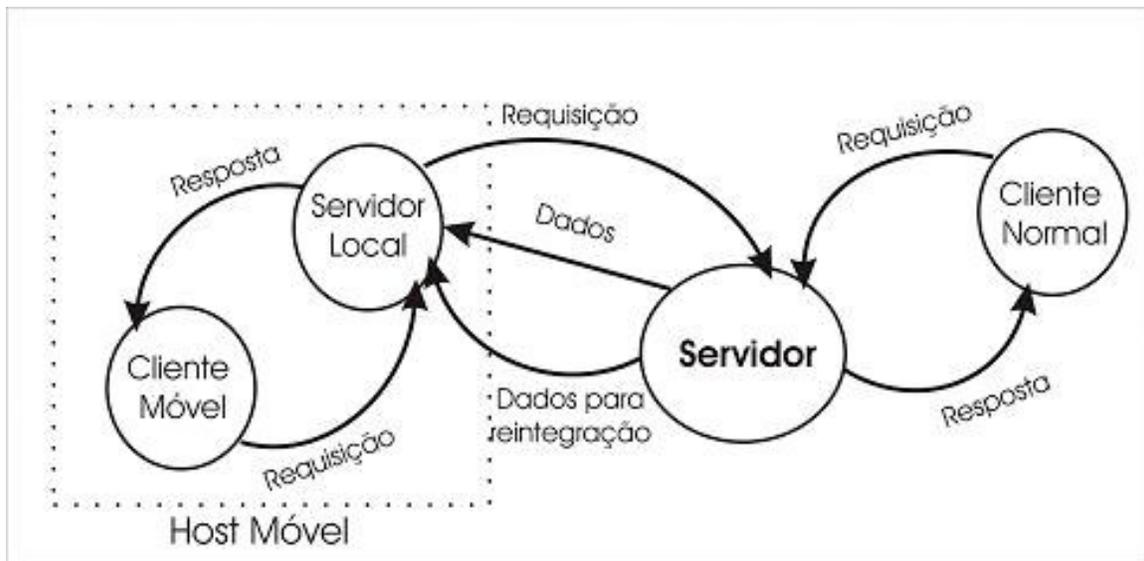


Figura 3.1. - Funcionamento da arquitetura de bancos de dados móveis

Nesse sentido, a recuperação de falhas e a reintegração de dados alterados localmente por clientes móveis em períodos de desconexão são de suma importância para garantir a confiabilidade de dados em sistemas móveis. Entretanto, o tratamento para tais operações foge do escopo deste trabalho, visto que tratamos aqui de consultas somente de leitura, as quais não afetam as informações armazenadas nas bases de dados.

3.1.3. Gestão de *Buffer* e *Cache*

Em relação à gestão de *buffer*, sabe-se que assim como um sistema cliente-servidor tradicional, os clientes podem administrar um *buffer* de páginas ou de objetos. Se os clientes têm *buffer* de páginas, então páginas inteiras são lidas ou gravadas no servidor toda vez que ocorre uma falha de página ou que uma página é esvaziada. Enquanto os *buffers* de objetos podem ler e/ou gravar objetos individuais e permitem o acesso a aplicativos objeto por objeto [64]. A manipulação desses dados no servidor móvel tem como unidade de acesso, em seu cache, itens de dados XML, ao invés de páginas. As razões pelas quais essa escolha foi feita são descritas de forma detalhada no capítulo 5.

Como discutido no capítulo anterior, no gerenciamento de cache em sistemas de bancos de dados móveis, outros fatores (além daqueles já considerados em sistemas de bancos de dados

distribuídos tradicionais), característicos de ambientes móveis, tais como localização e direção do movimento, devem ser considerados, a fim de que permaneçam nessa área de memória os itens mais prováveis de serem acessados e, ainda, que tais itens sejam confiáveis.

Em sistemas de bancos de dados distribuídos tradicionais, o estudo da consistência de cache está estritamente acoplado ao estudo do controle da concorrência e à replicação de dados, pois os dados colocados no cache podem ser acessados em diversos locais e de forma concorrente por vários clientes, e os bloqueios também podem ser inseridos no cache juntamente com dados nos clientes. Em sistemas de bancos de dados móveis, porém, a consistência pode ser comprometida também, devido à propriedade de dependência de localização. Dessa forma, a garantia de confiabilidade dos dados em cache em sistemas de bancos de dados móveis deve considerar dois fatores:

- Os dados colocados no cache podem ser acessados de forma concorrente por vários clientes;
- Os clientes possuem capacidade de locomoção.

Como descreveremos detalhadamente mais adiante, no Capítulo 5, nossa proposta aborda inicialmente somente o segundo fator, pois o mecanismo gerenciador de cache proposto trata apenas de consultas somente de leitura.

3.2. XML

Como introduzido no Capítulo 1, as informações no gerenciador de cache do servidor móvel são manipuladas no formato XML (*Extensible Markup Language*). Logo, consideramos necessária uma visão geral de alguns conceitos XML para fundamentar algumas escolhas do trabalho, que serão discutidas no decorrer do texto.

3.2.1. Introdução

XML é uma linguagem de marcação estrutural e semântica utilizada para descrever informações, especificado pela W3C. É muito popular na representação de dados na *Web* e vem ganhando popularidade também no armazenamento e na troca de dados em sistemas computacionais de maneira geral.

XML é um padrão completamente aberto, cujos documentos podem ser usados e reusados de diferentes formas e em diferentes formatos. Documentos XML são banco de dados de informações e apresentam-se legíveis e razoavelmente claros.

A tendência cada vez mais forte em torno da computação móvel e a crescente popularidade de XML têm resultado em mais aplicações de dispositivos móveis aceitando dados no formato XML. Em virtude disso, algumas empresas têm criado sistemas de gerenciamento de bancos de dados XML para aplicações empresariais, por exemplo, automação de vendas. Outros usam XML para definir protocolos de sincronização entre os servidores de bancos de dados globais e os bancos de dados móveis [40].

Os dados no servidor móvel são tratados e armazenados no formato XML. O acesso a essas informações ocorre através de padrões relacionados, tais como XSLT, DOM e SAX, como forma de facilitar a manipulação dos dados XML.

3.2.2. DOM

Os bancos de dados XML nativo (como Timber [29], Natix [35] e FoX [51]) são projetados especificamente para armazenar e manipular dados XML e utilizam padrões de acesso a dados, como por exemplo, o DOM.

Como uma especificação da W3C (*World Wide Web Consortium*), o objetivo do DOM (*Document Object Model*) [66] é fornecer uma interface de programação padrão para uma grande variedade de aplicações. Com o DOM, o desenvolvedor pode criar documentos XML, navegar em sua estrutura e adicionar, modificar ou excluir itens nessa estrutura.

O DOM é um padrão utilizado para criar uma estrutura de árvore a partir de arquivos XML e é uma API independente de linguagem e plataforma. Nesse sentido, um conjunto de interfaces foi padronizado para manipular o conteúdo dessas árvores e, dessa forma, as aplicações são capazes de manipular as árvores DOM como se estivessem manipulando o arquivo XML de origem.

As árvores são geradas por um *parser*, que consiste em um programa para criar uma estrutura de árvore a partir de um documento XML. A Figura 3.2 a seguir ilustra como ocorre esse procedimento. Um documento XML passa pelo *parser* e este cria uma estrutura de árvore DOM correspondente a este documento, que será manipulada pela aplicação.

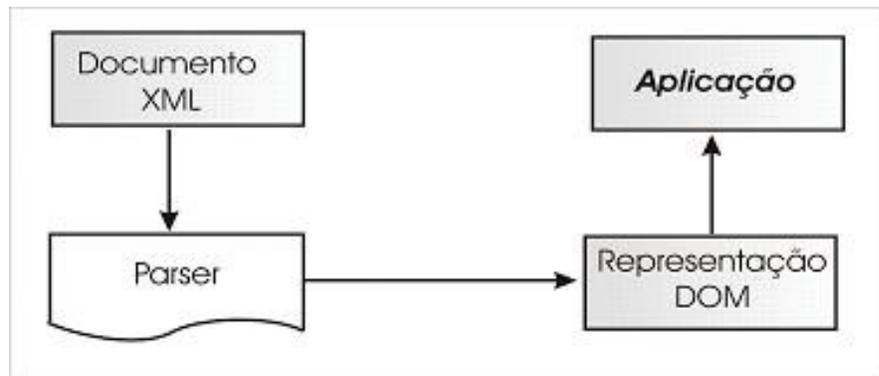


Figura 3.2 - Geração de árvore DOM a partir de documento XML

Como detalharemos em capítulos subseqüentes, este trabalho utiliza um *parser* denominado Xerces [1], que realiza a procura de itens em cache através da utilização do DOM. O DOM utiliza uma abordagem orientada a objetos e representa o documento como um conjunto de nós, onde cada nó descreve um tipo de informação presente no documento:

- Nó de Documento (*Document Node*);
- Nó de Elemento (*Element Node*);
- Nó de Atributo (*Attribute Node*);
- Nó de Texto (*Text Node*);
- Nó de Comentário (*Comment Node*).

Como exemplo, considere o documento `Livraria.xml`, mostrado no código a seguir (Figura 3.3).

```

<Livraria>
<Livro>
  <Titulo>Sistemas de Bancos de Dados Distribuidos</Titulo>
  <Autor>Vanduriez</Autor>
  <Preco>65,00</Preco>
</Livro>
<Livro Ano="1985">
  <Titulo>Sistemas Operacionais Modernos</Titulo>
  <Autor>Tanembaum</Autor>
  <Preco>102,00</Preco>
</Livro>
<Livro Ano="2000">
  <Titulo>Second Generation Mobile e Wireless Networks</Titulo>
  <Autor>U. Black</Autor>
  <Preco>112,00</Preco>
</Livro>
</Livraria>
  
```

Figura 3.3 - Documento `Livraria.xml`

Esse documento pode ser representado no formato de árvore, como forma de facilitar a manipulação dos itens do documento. Essa representação é ilustrada na Figura 3.4, onde os nós de texto contêm o conteúdo dos nós de elementos. Assim, para o quadrado relacionado ao atributo “Ano” do segundo elemento “Livro” na árvore, tem-se que seu valor é 1985 e o elemento “Título” deste mesmo nó tem como valor “Sistemas Operacionais Modernos”.

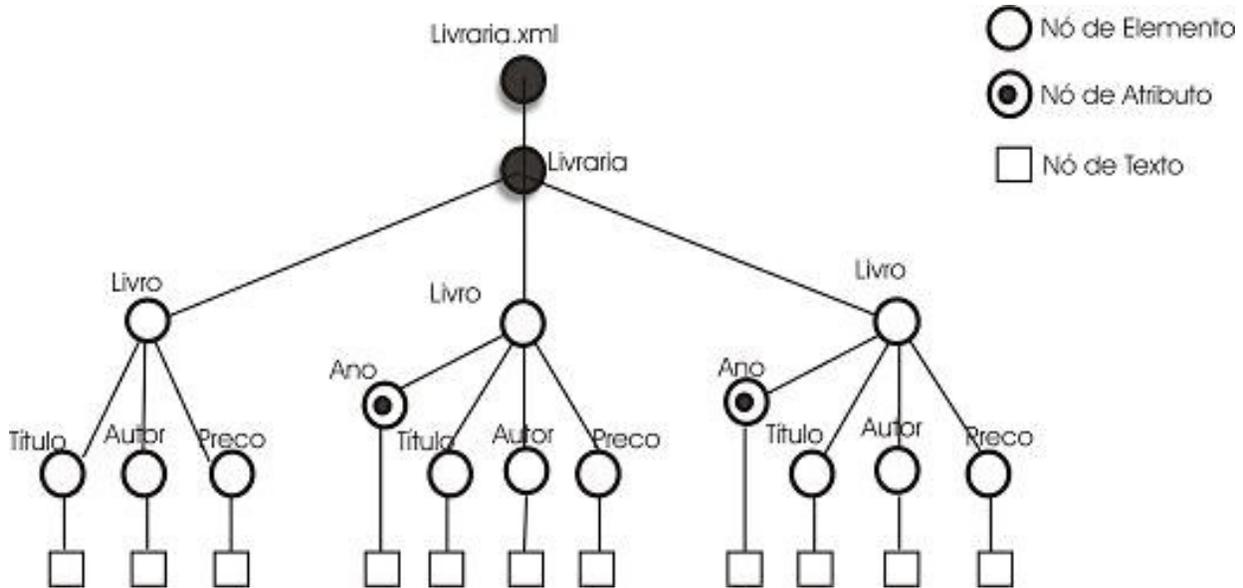


Figura 3.4. - Representação em DOM de Livraria.xml

3.2.3. Expressões de caminho (*XPath*)

Os SGBDs relacionais se beneficiam da definição inicial de um modelo de consulta preciso e formal e de um conjunto de primitivas algébricas universalmente aceitas [64]. Entretanto, com consultas baseadas em itens XML torna-se mais complicada uma definição de uma linguagem de consulta padrão, pois os dados XML não requerem uma estrutura rígida.

Felizmente há formas de se consultar dados no formato XML de forma eficiente, caracterizadas pela capacidade de navegação em árvores. As expressões de caminho de dados XML (*XPath*) constitui uma dessas formas, mapeando partes de um documento XML, através da utilização de uma sintaxe compacta.

XPath é uma linguagem utilizada para referenciar partes de um documento XML. Esta linguagem opera na estrutura lógica do documento (árvore) e explora o contexto hierárquico e seqüencial de elementos em um documento XML para localizar um elemento de interesse.

O servidor móvel, no qual atua o mecanismo de gerência de cache aqui proposto, recebe solicitações de dados dos usuários móveis através de um conjunto de expressões de caminho e, a partir desse conjunto, o gerenciador realiza a busca hierárquica dos dados disponíveis em cache, que são adequados para compor a resposta a essa requisição.

Uma expressão de caminho percorre a estrutura de dados XML para obter informação, ou formular uma condição sobre um atributo.

XPath utiliza expressões: *strings* com símbolos significativos, instruções para selecionar um elemento, atributo e seqüências de texto. Com o uso dessas expressões, os itens XML são identificados através de operadores especiais utilizados no *XPath*, dentre eles: o operador filho (*/*), que seleciona do filho imediato da coleção do lado esquerdo, o operador (*//*), que seleciona dos descendentes arbitrários da coleção do lado esquerdo e o operador de atributo (*@*), que identifica um atributo de um item XML.

Como forma de esclarecer a utilização dessas expressões, consideramos o exemplo apresentado na seção anterior, onde a Figura 3.4 representa os nós gerados a partir de um documento XML. Temos, dessa maneira, que a expressão de caminho *Livraria/Livro/Autor* representa o elemento autor do objeto livro, enquanto *Livraria/Livro/@Ano* representa o atributo ano de publicação do objeto livro.

3.2.4. Dados XML em ambientes móveis sem fio

A popularidade de XML deve-se, em grande parte, à sua flexibilidade para representar quaisquer tipos de informação. A utilização de *tags* permite uma fácil descrição dos dados e, além disso, a natureza extensível do XML torna possível definir novos tipos de documentos para propósitos específicos.

Essas características de XML tornam a sua utilização bastante adequada em ambientes móveis sem fio. A quantidade de informação que trafega nas redes de comunicação sem fio, sejam elas de longa ou de curta distância, cresce a cada dia e o ritmo acelerado no qual estes dados devem ser processados são fatores que estimulam a utilização de XML como formato padrão para troca e manipulação de dados em sistemas móveis.

É importante considerar, no entanto, que outros mecanismos eficientes de armazenamento de dados são também largamente utilizados atualmente em ambientes móveis sem fio, como SGBDs relacionais adaptados para dispositivos portáteis. Esses modelos de gerenciamento de dados não invalidam a utilização de XML como formato para manipulação e troca de dados através de redes móveis, sendo necessárias, porém, conversões entre formatos de dados.

3.3. Conclusão

Neste capítulo, foram apresentadas características de sistemas de bancos de dados móveis e introduzidos conceitos de XML, que será utilizado para tratamento e transferências de dados no mecanismo de gerência de cache proposto.

No capítulo seguinte, estudamos diversas soluções em gerência de cache, de modo a apresentar o estado da arte desta dissertação e discutir as contribuições adicionais deste trabalho em relação aos demais encontrados na literatura.

CAPÍTULO 4

SOLUÇÕES EM GERÊNCIA DE CACHE

Neste capítulo são descritos trabalhos que se relacionam com a proposta aqui apresentada e que serviram de base para a nossa pesquisa. Esses trabalhos concentram-se na área de gerência de cache e na área de ambientes móveis sem fio. Na seção 4.1 são apresentados trabalhos referentes à gerência de cache em sistemas distribuídos, na seção 4.2 são estudados vários trabalhos na área de gerência de cache em sistemas móveis sem fio, de suma importância para fundamentar esta dissertação. Na seção 4.3 são citadas publicações sobre gerência de cache utilizando XML e, finalmente, a seção 4.5 mostra uma breve discussão sobre os estudos descritos neste capítulo e os problemas não abordados por estes estudos e que pretendemos solucionar nesta dissertação.

4.1. Gerência de Cache em Sistemas Distribuídos

A grande popularidade de sistemas de banco de dados cliente-servidor contribuiu para o surgimento de várias pesquisas [17][19][20][64], com o intuito de melhorar o desempenho desses sistemas. Dentre essas pesquisas, estão as contribuições referentes à gerência de cache nesse tipo de ambiente.

Em muitos trabalhos encontrados na literatura, a manutenção de dados no cache do cliente reduz a necessidade de se obter dados a partir de serviços ou outros pontos na rede. A gerência da área de memória reservada para cache deve garantir que os dados nela mantidos sejam os mais prováveis de serem acessados no futuro e deve, ainda, assegurar que tais dados são confiáveis.

Baseados nessas premissas, vários trabalhos apresentam contribuições de gerência eficiente de cache no cliente. Como, por exemplo, em [17][19] onde são apresentados algoritmos de manutenção de consistência do cache e uma investigação sobre o desempenho de tais algoritmos. Ou ainda, como em [20] onde são analisados algoritmos de consistência em uma extensão de cache em memória – o cache

em disco, ou seja, são investigados os impactos de se manter um grande volume de dados consistentes em um cache de disco.

Apesar do fato de sistemas móveis serem uma extensão de sistemas distribuídos, os trabalhos apresentados para gerência de cache em sistemas de bancos de dados distribuídos não são suficientes para a criação de mecanismos eficientes de gerência de cache em ambientes móveis. Essa restrição deve-se, dentre outros motivos, ao fato de que em ambientes móveis a característica de possibilidade de locomoção dos usuários móveis afeta esses meios de gerenciamento e, por isso, essa característica deve ser considerada. Nesse sentido, pesquisas referentes à gerência de cache em sistemas móveis têm sido desenvolvidas com o intuito de prover um serviço mais adequado a esses sistemas.

4.2. Gerência de Cache em Ambientes Móveis

Os constantes avanços em tecnologias de computadores portáteis, celulares e PDA's (*Personal Digital Assistants*) adicionados à crescente utilização desses dispositivos têm servido como forte incentivo para pesquisas na área de sistemas móveis sem fio [3][4][14][15][18][31][43][45][7], como forma de prover, cada vez mais, serviços de comunicação mais confiáveis e menos onerosos.

A mobilidade tem aberto novas áreas de pesquisa em sistemas de gerenciamento de bancos de dados distribuídos e de redes, pois técnicas tradicionais desenvolvidas para esses sistemas têm sido desenvolvidas assumindo-se que a localização dos usuários e as conexões entre eles não se modificam. Em um ambiente de computação móvel, usuários carregam computadores portáteis e desejam manter acesso transparente à rede através de *links* sem fio enquanto se movimentam de um lugar para outro [22]. Espera-se ainda que, em um futuro próximo, milhões de usuários móveis façam uso de aplicações integradas de voz, dados e imagens [52].

Características e desafios da computação móvel são apresentados em [18][4][43]. Como principais características dos ambientes móveis, podem ser citadas a comunicação sem fio, a mobilidade e a portabilidade [43]. Tais características fazem com que o gerenciamento de sistemas móveis sem fio se torne mais complexo do que em sistemas distribuídos tradicionais. Dessa forma, vários trabalhos têm sido apresentados com o intuito de fornecer suporte adequado a sistemas móveis, levando em consideração as características acima mencionadas.

Entre esses trabalhos estão incluídas pesquisas como arquitetura, segurança e desafios de bancos de dados móveis [65][4][3], processamento de consultas e gerência de transações levando em consideração a característica de dependência de localização dos ambientes móveis [15][31][36][41][43][22][27] e gerência de cache em sistemas móveis sem fio [8][44][45][55][56][57][70][72].

Este último constitui o foco do nosso trabalho e possui um grande número de contribuições. Um mecanismo eficiente de gerência de cache pode melhorar consideravelmente o desempenho de sistemas móveis, pois é possível diminuir o fluxo de dados entre cliente e servidor através da interface aérea. Este fato tem contribuído para que a gerência de cache em sistemas móveis sem fio seja um tema de crescente interesse tanto no meio científico[34][45][55], quanto no campo comercial [62].

Diversos trabalhos nesta área, referentes às políticas de substituição e métodos de validação no cliente móvel, têm sido apresentados, no intuito de contribuir para uma melhor utilização do *link* aéreo, visto que este é um recurso caro e um meio lento e pouco seguro de transporte de dados em ambientes móveis sem fio.

Muitos trabalhos propõem mecanismos adaptativos de gerência de cache para sistemas móveis, nos quais características específicas destes sistemas (localização do dispositivo e/ou direção de movimento) são fatores relevantes na gestão de cache. Em [45], por exemplo, o mecanismo proposto não é baseado em páginas, visto que mecanismos de gerência de cache baseado em páginas não são adequados para ambientes móveis sem fio, pois dados desnecessários podem ser transferidos em páginas, aumentando, desse modo, a utilização do *link* aéreo e os custos de comunicação. A estratégia de substituição adotada em [45] é baseada em padrões de acesso dos itens, ao invés dos comumente empregados LRU e MRU. Em relação às estratégias de validação, cada cliente móvel é responsável por validar seus itens em cache, através de mensagens vindas do servidor, desse modo, somente a versão mais recente pode ser acessada.

Esse mecanismo é construído no nível de modelo de dados e uma tabela de cache é mantida pelo cliente móvel para indicar se o item de dado está armazenado localmente ou reside no servidor. Tal mecanismo de gerência de cache de alto nível foi proposto por [45] principalmente porque uma tabela de cache é simplesmente um mini banco de dados e pode ser gerenciado pelo SGBD.

Tal mecanismo, no entanto, não é utilizado em qualquer tipo de banco de dados, pois utiliza um subconjunto de construtores utilizados na modelagem OODB. Dessa forma, para que o gerenciamento de cache proposto em [45] funcione é necessário que clientes e servidores possuam gerenciamento de banco de dados orientado a objetos e que executem consultas utilizando OQL.

Um modelo de mobilidade é desenvolvido em [55] para representar o comportamento de usuários móveis, além de definir formalmente consultas dependentes de localização e investigar estratégias de processamento de consultas e gerenciamento de cache. A política de substituição de cache desenvolvida no trabalho é denominada FAR (*Furthest Away Replacement*), a qual escolhe os itens a serem retirados do cache de acordo com o status do usuário. A determinação deste status é baseada na localização e velocidade do dispositivo.

Outros mecanismos adaptativos propostos são baseados em *caching semântico* [44][55][56], onde informações sobre o dispositivo são adicionadas aos itens armazenados para serem levadas em consideração no processo de substituição em cache. A idéia de *caching semântico* é manter descrições semânticas e respostas associadas de consultas prévias no cliente móvel.

Uma extensão de *caching semântico* é apresentado em [57], onde é proposto um método de agrupamento (*clustering*) em cache semântico, no qual resultados de consultas relacionados semanticamente (com tuplas resultantes em comum) são agrupados em cache. O método de substituição proposto é o *LRU de dois níveis*, o qual faz uso das propriedades de localidade temporal, baseado no tempo decorrido desde que o item foi acessado pela última vez e na localidade semântica, que se baseia em padrões específicos de acesso, tais como localização e direção de movimento. Baseado na localidade semântica, sabe-se que se uma parte do *cluster* foi visitada recentemente, as outras partes são prováveis de serem acessadas também. Assim, um *cluster* é tratado como um todo na escolha dos segmentos a serem substituídos.

Nenhum trabalho sobre *caching semântico*, porém, admite a possibilidade de determinado valor de dado em cache ser válido em uma localização diferente daquela associada à consulta. Para contornar tal problema pode ser utilizada a idéia de *escopo válido*. O escopo válido de um valor de dado é definido como a área geográfica na qual este valor é válido.

Um método de validação e estratégias de substituição em cache baseado no uso do escopo válido é proposto em [72]. A estratégia de substituição apresentada neste trabalho consiste, simplesmente, em eliminar o elemento de menor escopo válido. O método de validação consiste em anexar informações de validação parciais ou completas aos valores dos dados. Assim, quando um valor de dado é entregue do servidor para o cliente, seu escopo válido é entregue juntamente a este valor, permitindo ao cliente checar a validade dos dados.

Em geral, há dois tipos de estratégias de validação de cache, os métodos iniciados por cliente e os métodos iniciados por servidor. No método iniciado por cliente, o cliente monitora os estados dos

itens em cache e inicia procedimentos de checagem de validade, enquanto no método iniciado por servidor, o servidor monitora os estados dos itens em cache e informa ao cliente para que descarte dados obsoletos [70].

Os métodos iniciados pelo servidor são uma estratégia eficiente para validação baseada em atualização de dados [5][32]. Neste método, relatórios de validação são enviados pelo servidor e, se estiver ativo, cada cliente móvel verifica esses relatórios e atualiza seu cache. Em algoritmos adaptativos de validação de cache, encontrados na literatura (tais como [67][70]), o próximo relatório é decidido dinamicamente baseado no fluxo de atualização de dados.

Relatórios de validação imediatos, enviados constantemente por um servidor, não parecem ser um método adequado para validar dados em ambientes móveis [45], porque cada dispositivo se conecta e desconecta frequentemente na rede sem fio, além do aumento de utilização da interface aérea devido ao envio constante desses relatórios.

Os métodos iniciados pelo cliente são usados em validações dependentes de localização, ou seja, em validação baseada em mudança de localização do usuário [70]. Eles são, por esse motivo, a base das estratégias adotadas para manutenção de coerência de cache no servidor de dados móvel do nosso trabalho.

Vários métodos de validação de cache em clientes móveis baseados em *relatórios de validação (Invalidation Report - IR)* têm sido propostos, como em [8], [71] e [5]. Nestes métodos, o servidor distribui, periodicamente a todos os clientes, um relatório no qual os itens de dados modificados são indicados, ou seja, ao invés de consultar o servidor a respeito da validade dos itens armazenados, os clientes podem “escutar” esses IRs para validar seu cache. Porém, esse mecanismo de distribuição de IR pelo servidor pode se tornar ineficiente se a periodicidade de envio for pequena, pois o custo de transmissão em redes móveis sem fio é alto.

Os trabalhos pesquisados na área de gerência de cache em ambientes móveis propõem um gerenciamento de cache na memória do cliente e, em nenhum trabalho pesquisado, foram encontradas melhorias no que se refere às estratégias de cache em redes *ad hoc*. As contribuições encontradas hoje na literatura são, em geral, eficientes para clientes móveis. No entanto, no que se refere à utilização de redes móveis sem fio de curta distância não parecem ser adequadas, pois não são capazes de aproveitar os benefícios e facilidades que essas redes oferecem e, além disso, não provêm soluções para as limitações dessas redes.

4.3. Gerência de Cache utilizando XML

Baseado na proliferação recente de dados XML e na emergência da linguagem XQuery, propostas de *frameworks* de processamento de consultas XQuery [16] e sistemas de gerenciamento de cache baseados em XML têm sido apresentadas nos últimos anos, com o intuito de armazenar em cache dados no formato XML para agilizar ainda mais a resposta às consultas.

O ACE-XC [9], por exemplo, é um sistema de cache para consultas XQuery que aplica modos de armazenamento e estratégias de reescrita de consultas para responder a novas consultas baseadas nos resultados em cache. Um cálculo estatístico é utilizado na estratégia de substituição dos itens em memória. Este cálculo é baseado na frequência de acesso e no espaço que será liberado por um dado item.

O Xcache é brevemente apresentado em [10]. Neste sistema os autores propõem a utilização de *caching* semântico para sistemas de processamento de consultas XML. Entretanto, a utilização de semântica de dados a ser utilizada para armazenar dados em cache é apresentada de forma genérica, não apresentando casos específicos, tais como consultas dependentes de localização.

Como ilustrado na Figura 4.1, ambos os trabalhos sobre gerenciamento de cache baseado em XML apresentam sistemas cuja arquitetura consiste basicamente em dois subsistemas: o Manipulador de Consultas, que implementa o repositório e a reescrita de consultas, e o Gerenciador de Cache, que gerencia o espaço do cache e aplica técnicas de armazenamento e substituição. Nesta figura são apresentados os principais módulos que compõem o *framework*. Entre eles, estão:

- O analisador de consultas (*Query Decomposer*) que aplica regras de normalização a uma entrada XQuery de forma a derivar uma forma normalizada, de acordo com os padrões de consulta estabelecidos;
- O registrador de padrão de consulta (*Query Pattern Register*) que utiliza variáveis padrões e seus relacionamentos de dependências para listar as consultas XQuery em regiões semânticas;
- O projetor de restrições de consulta (*Query Containment Mapper*) que estabelece o mapeamento de restrições de consultas para cada nova consulta;

- O escritor de consultas (*Query Rewriter*) que reescreve a consulta de acordo com os mapeamentos estabelecidos no projetor de restrições de consulta;
- O gerenciador de substituição (*Replacement Manager*) que incorpora estratégias de substituição, para liberar espaços de memória para novas consultas;
- O módulo responsável por combinar as regiões (*Region Coalescer*) que aplica técnicas de fusão de espaços de memória, para controlar a granularidade das regiões utilizadas para armazenar os dados.

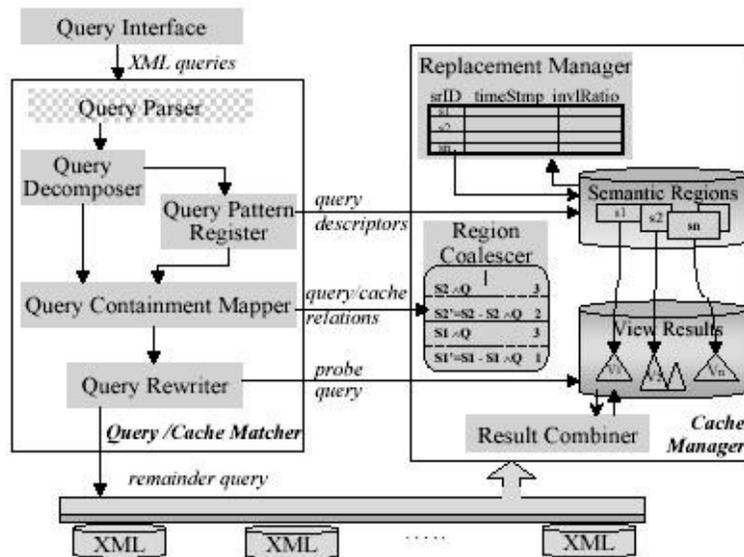


Figura 4.1 – Arquitetura Sistema Gerenciador de Cache baseado em XML²

4.4. Discussão

Nenhum dos trabalhos mencionados na seção 4.3 considera particularidades de redes móveis sem fio de curta distância, como por exemplo, a possibilidade de que resultados de consultas possam ser reutilizados por outros clientes em uma rede *ad hoc*. Se alguns clientes executam consultas semelhantes, há então, a possibilidade de existirem vários dispositivos armazenando resultados que poderiam ser aproveitados para outros usuários naquela rede. Além disso, nenhum dos trabalhos

2. Adaptado de [9] e [10]

armazena itens XML em bancos de dados adaptados para ambientes móveis sem fio, ou seja, as estruturas de armazenamento em cache para sistemas móveis não são adaptadas para prover acesso e recuperação de dados no formato XML. No que se refere aos sistemas de gerência de cache baseados em XML, como o XCache [10], não constam trabalhos que considerem casos específicos para ambientes móveis sem fio.

Nosso trabalho se diferencia dos demais na área de gerência de cache para tecnologia móvel sem fio devido ao fato de que o nosso foco está na gestão de cache em um servidor de dados móvel e não no cliente. Ou seja, um equipamento portátil será capaz de realizar funções de gestão de cache em XML nativo, pré-processamento de consultas, dentre outras, servindo a outros dispositivos numa rede sem fio local.

4.5. Conclusão

Neste capítulo, mostramos diversas soluções em gerência de cache encontradas na literatura e introduzimos as contribuições adicionais apresentadas neste trabalho, com uma breve descrição comparativa.

No capítulo seguinte, apresentamos, de forma minuciosa, a descrição do mecanismo de gerenciamento de cache proposto, onde são apresentadas as estratégias implementadas neste mecanismo e a forma na qual tais estratégias atuam em um ambiente móvel sem fio.

CAPÍTULO 5

GERENCIADOR DE CACHE PARA UM SERVIDOR DE DADOS MÓVEL

Neste capítulo são apresentadas as principais características do mecanismo de cache proposto e as estratégias de gerenciamento utilizadas nesta dissertação, para que o gerenciador de cache de dados em um servidor móvel seja adaptado eficientemente para o ambiente sem fio. Na Seção 5.1 é apresentada a arquitetura considerada no trabalho, na Seção 5.2 são descritas as características e funcionalidades do servidor móvel. Na Seção 5.3. apresentamos, de forma detalhada, as estratégias de substituição e validação utilizadas no mecanismo de gerenciamento de cache proposto e, finalmente, na Seção 5.4, descrevemos sobre a manutenção dos dados em cache.

5.1. Introdução

Conforme discutido no Capítulo 4, muitas pesquisas sobre gerenciamento de cache em sistemas móveis sem fio têm sido realizadas com o intuito de melhorar o desempenho desses sistemas. No entanto, percebe-se uma carência de contribuições científicas que considerem características de um determinado tipo de rede. Por exemplo, com o crescimento na utilização das redes *ad hoc*, surge a necessidade de se fornecer um mecanismo de gerência de cache mais conveniente para esse tipo de rede móvel, de maneira que suas especificidades sejam consideradas na escolha do melhor mecanismo de gerência de cache a ser utilizado.

O mecanismo de gerenciamento de cache aqui apresentado é bastante adequado para redes *ad hoc*, visto que é possível dispor de um servidor móvel de dados que será capaz de gerenciar o fluxo de dados nessa rede. Como exposto no Capítulo 2, as redes sem fio de curta distância apresentam uma característica especial que é a possibilidade de mudanças de localização tanto entre os dispositivos que compõem a rede, quanto da própria rede como um todo. Dessa forma, essa característica é um ponto

importante considerado nesta dissertação para fornecermos um mecanismo adequado para gerenciamento de cache nesse tipo de rede.

Desse modo, essa dissertação visa complementar os trabalhos já existentes na área de gerência de cache para ambientes móveis sem fio, no sentido de aumentar o aproveitamento dos dados que trafegam pelas redes móveis.

Neste capítulo, são introduzidas as estratégias utilizadas pelo gerenciador de cache do servidor móvel. Porém, para a demonstração formal de algumas dessas estratégias, é necessário conhecer as estruturas de manipulação necessárias para o gerenciamento de cache e, por esse motivo, alguns algoritmos são apresentados e discutidos apenas no Capítulo 6.

5.2. Arquitetura

Desde o surgimento de sistemas móveis sem fio, vários meios para a comunicação de dispositivos portáteis foram desenvolvidos. Um modelo largamente aceito para sistemas móveis é a rede celular [22]. O modelo celular, como discutido no Capítulo 2, é utilizado nas redes sem fio de longa distância. Há, ainda, modelos específicos para redes sem fio de curta distância (redes *ad hoc*), que oferecem comunicação mais barata e atendem, de modo satisfatório, a vários propósitos específicos, tais como grupos virtuais, onde usuários com interesse em comum mantêm comunicação.

A arquitetura considerada no presente trabalho consiste em dois tipos de rede, uma rede móvel de curta distância, como o *Bluetooth* ou IEEE 802.11 e uma rede móvel de longa distância, como, por exemplo, um sistema celular GPRS.

Um equipamento portátil (e.g. *laptop*) atua como servidor de outros dispositivos móveis (e.g. *palms*, celulares) na rede *ad hoc*, porém continua a ser cliente de um servidor fixo (ou externo) na rede sem fio de longa distância que disponibiliza mecanismos para que o servidor móvel atualize seu cache. A comunicação entre esses dois servidores, assim como de qualquer dispositivo móvel com o servidor fixo, é feita através da interface aérea da rede celular.

No modelo de arquitetura ilustrado na Figura 5.1, os dispositivos portáteis da rede móvel de curta distância fazem requisições para o servidor de dados móvel que as encaminha ao servidor fixo ou as responde com os dados em memória, porém os dispositivos móveis continuam aptos a se comunicarem com o servidor fixo de dados, em caso de falhas no servidor móvel.

O servidor móvel atua em um equipamento móvel de maior capacidade de armazenamento e processamento, possibilitando, desse modo, maior disponibilidade de dados na rede *ad hoc*.

Como quaisquer dispositivos portáteis em uma rede móvel sem fio, os usuários que utilizam a rede *ad hoc* são capazes de mudar de localização frequentemente dentro desta rede e a rede como um todo também pode ter sua posição alterada. Este fato acarreta algumas implicações no gerenciamento de comunicação, como por exemplo, impedir que os dispositivos sejam desconectados da rede de longa distância ao se locomoverem. Este processo de garantia de continuidade de conexão é garantido através de processos de *handoff* e foge do escopo do nosso trabalho, sendo que a mudança de localização é um importante fator considerado pelo servidor móvel na gerência de solicitações dos clientes da rede *ad hoc*, a qual detalharemos nas seções a seguir.

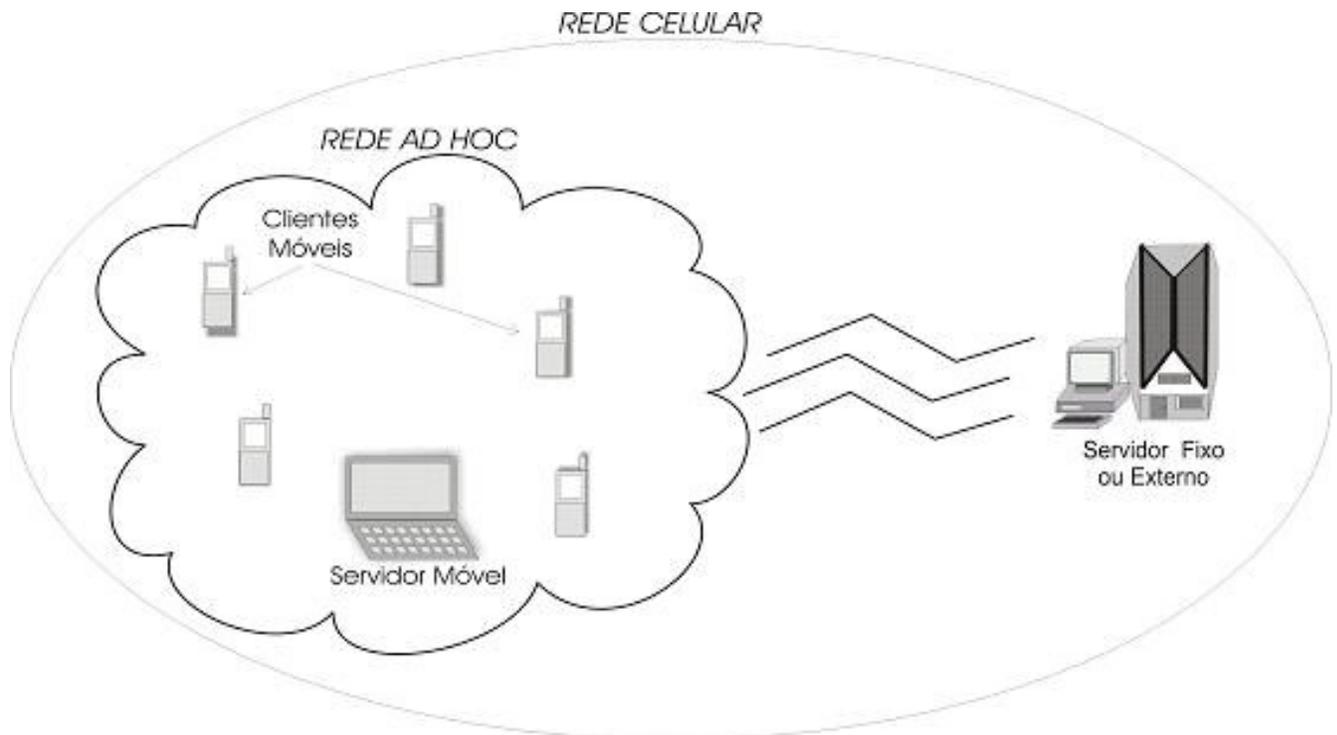


Figura 5.1 – Arquitetura Considerada

5.3. O Servidor Móvel

5.3.1. Características

Conforme introduzido no Capítulo 1, a utilização de redes sem fio de curta distância por grupos de usuários com interesses em comum tem aumentado consideravelmente. Nessas redes, os dispositivos que a compõem interagem entre si e fazem, freqüentemente, requisições a um servidor externo (não pertencente à rede *ad hoc*). Nesse sentido, a existência de um dispositivo utilizado como um servidor local nesta rede facilita a organização e a gerência das requisições acima mencionadas.

Esse servidor, que é também um dispositivo móvel, interage com os outros dispositivos da rede *ad hoc* (clientes) e com o servidor externo que fornece os dados requisitados. A principal finalidade do uso desse servidor móvel é, portanto, diminuir a comunicação entre os dispositivos da rede *ad hoc* com o servidor fixo (externo). Além disso, dispondo de um servidor móvel, com uma capacidade de armazenamento maior que a dos outros equipamentos portáteis, é possível aumentar a disponibilidade dos dados, obviamente por tempo limitado, em caso de desconexão com o servidor fixo da rede celular.

Os modelos de gerenciamento de cache para ambientes móveis sem fio apresentados na literatura até o presente momento propõem o armazenamento dos itens de maior probabilidade de acesso na memória do cliente [8][34][44][45][55][71]. Tais propostas visam, dentre outros benefícios, aumentar a disponibilidade de dados e diminuir o tráfego de dados entre cliente e servidor.

Neste trabalho propomos um gerenciamento de cache mais eficiente para uma rede *ad hoc*, o qual atua em um servidor de dados móvel. Por possuírem, geralmente, interesses em comum, é muito provável que os usuários dessa rede realizem requisições semelhantes, ou seja, alguns dados retornados em resposta a uma consulta de um usuário podem ser utilizados novamente para responder às requisições de outro(s) usuário(s).

Diante desse fato, um gerenciamento de cache no servidor móvel diminui ainda mais o tráfego de dados na interface aérea da rede de longa distância e aumenta a disponibilidade dos dados na rede *ad hoc* considerada.

5.3.2. Armazenamento FoX

O servidor fixo e o servidor móvel mencionados neste trabalho devem efetuar o armazenamento de informações utilizando o FoX [51], um projeto de Banco de Dados XML Nativo, independente de esquema, que vem sendo tema de várias dissertações em andamento na Universidade Federal do Ceará.

O FoX compreende dois sub-sistemas principais: o Processador de Consultas e o Gerenciador de Armazenamento (Figura 5.2). O Processador de Consultas é baseado na álgebra XML, publicada pela W3C e atua nas consultas, reconhecendo, otimizando e gerando os planos de execução, enquanto o Gerenciador de Armazenamento interage no nível mais baixo e é responsável pelo armazenamento e indexação do conteúdo XML. Esse gerenciador de armazenamento desempenha as funções de recuperação, inserção e remoção do conteúdo XML de acordo com as consultas e atualizações feitas sobre o mesmo na base de dados.

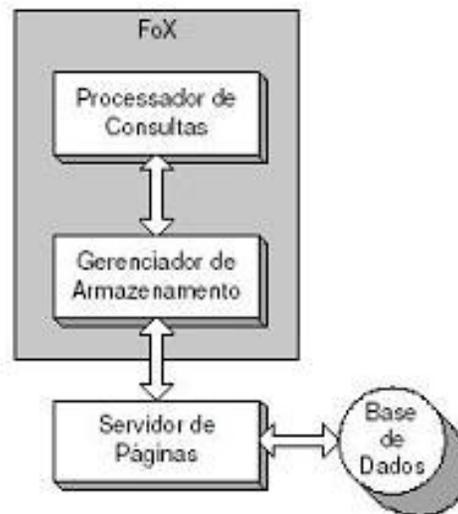


Figura 5.2 - Arquitetura Geral do FoX

É necessário, porém, adaptar o gerenciamento de cache do FoX para o ambiente móvel, onde passam a ser consideradas características desse tipo de ambiente, como por exemplo, localização geográfica. Essa adaptação consiste em um mecanismo de gerência de cache, no qual novos critérios de substituição e validação devem ser adicionados aos critérios já considerados em sistemas distribuídos tradicionais, pois no ambiente móvel, os usuários têm a capacidade de se locomover de uma posição à outra, fato este que influi diretamente no estado dos dados em cache.

Dessa forma, o gerenciador de cache no servidor móvel deve ser separado do gerenciador de armazenamento, atuando, assim, em um módulo intermediário, entre o processador de consultas do FoX e seu gerenciador de armazenamento, como ilustrado na Figura 5.3.



Figura 5.3 – Gerenciador de Cache no FoX, adaptado para Ambientes Móveis

O armazenamento de dados do FoX mantém-se inalterado para este trabalho, de maneira que não tratamos de pontos referentes ao armazenamento de dados no servidor móvel e nem às características de replicação por ele acarretadas. Dessa forma, neste trabalho, mantemos o foco somente dos mecanismos relacionados à gerência de cache do servidor móvel.

É importante perceber que essa atuação conjunta do gerenciador de cache proposto com o FoX visa integrar as funcionalidades de recuperação e tratamento de dados e processamento de consultas. Por realizarem tratamento de informações baseado em XML, essa integração ocorre de maneira natural, sem necessidades de conversão de dados.

5.3.3. Serviços de Dados

O servidor móvel deve servir aos clientes da rede *ad hoc*, fornecendo dados dos quais ele mesmo dispõe ou encaminhando uma solicitação ao servidor fixo, caso esses dados não estejam disponíveis localmente.

A solicitação de dados é feita através de expressões de caminho XML e as informações transmitidas e manipuladas pelo servidor móvel estão no formato XML.

Os clientes móveis enviam solicitações de dados ao servidor móvel e, ao receber uma consulta, o servidor móvel deve verificar se esta consulta pode ser respondida localmente, a fim de evitar encaminhar diretamente a solicitação ao servidor fixo.

No servidor móvel, a busca de itens de dados que irão compor a resposta a uma consulta submetida por clientes móveis ocorre como ilustrado no diagrama de atividades da Figura 5.4.

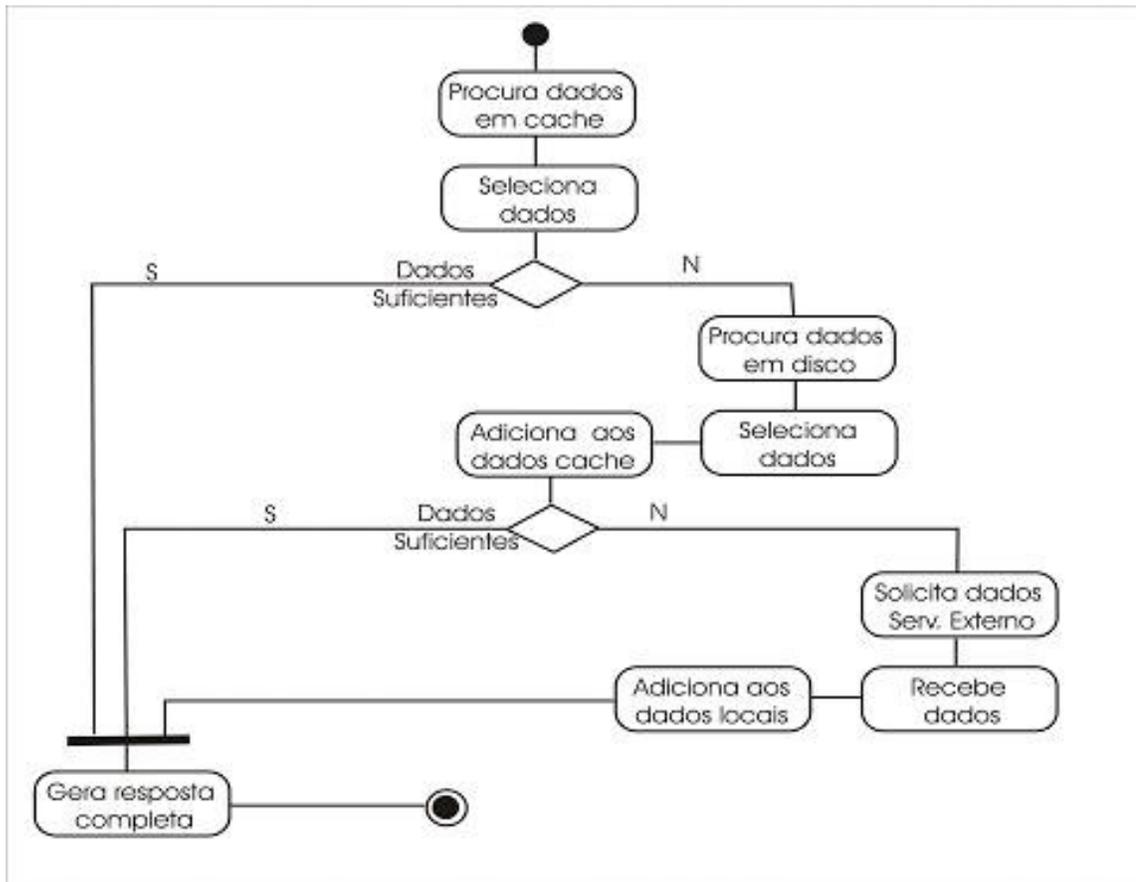


Figura 5.4 – Diagrama de Atividade referente à operação de busca de dados pelo servidor móvel de dados

Essa busca de dados é feita em dois níveis, onde o primeiro nível consiste no cache local e o segundo é representado pelo disco do servidor móvel. Assim, diante de uma solicitação de dados, o servidor móvel verifica se esses dados, ou pelo menos parte deles, estão em seu cache e, em seguida, realiza o mesmo procedimento de verificação em seu disco local. Caso os dados encontrados localmente não sejam suficientes para compor a resposta à consulta, o restante dos itens necessários é, então, solicitado ao servidor fixo, para finalmente, fornecer o resultado ao cliente móvel.

5.4. O Gerenciador de Cache

5.4.1. Tratamento de Consultas Dependentes de Localização

A capacidade de locomoção dos clientes em um ambiente móvel fez surgir a possibilidade de que respostas a algumas consultas variem de acordo com a posição corrente do usuário [4], adicionando, dessa forma, novos conceitos e problemas, não presentes em sistemas distribuídos tradicionais.

Dentre esses novos conceitos estão:

- *Dado Dependente de Localização (DDL)* - dado cujo valor é determinado pela localização para a qual ele é relacionado;
- *Consulta Dependente de Localização (CDL)* - uma consulta que é processada a partir de DDL's, e cujo resultado depende do critério de localização explícito ou implicitamente especificado. Além disso, seu resultado pode se modificar quando o usuário muda de posição [55].

O conceito de consultas com critérios de localização foi apresentado também em [27], onde são introduzidos alguns exemplos de consultas de usuários móveis, desde consultas mais típicas, tais como “Onde é X” ou “Onde é o hospital mais próximo”, até aquelas mais complexas, como “Retorne a melhor rota para o hospital mais próximo com as melhores condições de tráfego”. O foco principal no processamento dessas consultas é minimizar o custo de comunicação para recuperar as informações necessárias para responder à consulta e mantê-las consistentes.

Estratégias utilizadas para tal problema podem ser simples, resultando em um grande número de mensagens e longos períodos de latência, entretanto podem ser extremamente complexas, visto que construir planos de processamento ótimos pode chegar a ser um problema NP-Completo [4].

Como forma de diminuir o custo de comunicação na recuperação de dados em resposta às CDL's, propomos um mecanismo de gerenciamento de cache, no qual são armazenados resultados de consultas prévias. Esse gerenciamento é realizado a fim de que os dados mais prováveis de serem requisitados no futuro estejam disponíveis e sejam confiáveis, diminuindo, dessa forma, as requisições para o servidor fixo da rede sem fio de longa distância. Vale lembrar que as consultas consideradas nesse trabalho são consultas somente de leitura, o que implica que não tratamos aqui de nenhum tipo de controle de concorrência de dados.

Consideremos o exemplo a seguir, como forma de esclarecer o processamento de CDL's. O cenário deste exemplo consiste em usuários de uma rede *ad hoc*, onde cada dispositivo desses usuários é identificado por d_n (vide Figura 5.5).

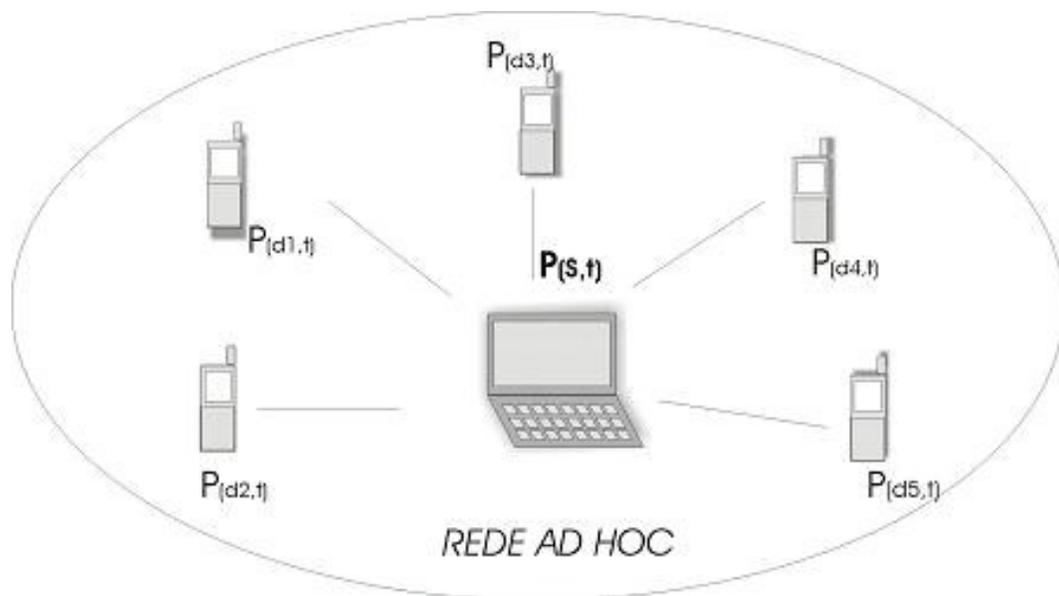


Figura 5.5 - Posições dos dispositivos, utilizadas no processamento de CDL's

Seja $P(S,t)$ a posição P do servidor móvel S na rede *ad hoc* em um momento t . Tal posição é utilizada na determinação da distância dos itens (fornecidos como resposta a uma determinada requisição) aos dispositivos dessa rede, distância esta considerada como fator mais crítico nos procedimentos de substituição em cache, conforme discutiremos mais adiante e, seja finalmente, $P(d_n, t)$ a posição P do dispositivo d_n no instante de tempo t . Vale ressaltar que as posições citadas em todo o trabalho são dadas através de coordenadas geográficas.

Os usuários estão, inicialmente, em uma rede *ad hoc* de posição P' , e o usuário do dispositivo d_1 submete a seguinte consulta: “Retorne os restaurantes mais próximos a um raio de 5 km.”. O resultado – restaurantes A, B, C e D, mostrados na Figura 5.6 - é enviado como resposta à essa consulta.

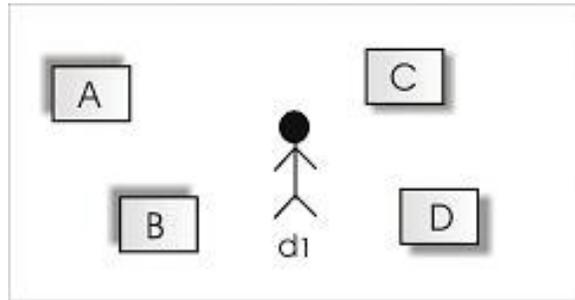


Figura 5.6 - Resultado da consulta submetida pelo usuário d_1 em posição inicial

O grupo como um todo (conectado pela rede *ad hoc*) se movimenta então para a área P'' , distante 5 km de P' (vide Figura 5.7). Na nova posição, um outro usuário de dispositivo d_2 , submete a mesma consulta citada anteriormente e, como podemos observar, o resultado a ser retornado ao cliente consiste nos restaurantes C, D, E, F e G.

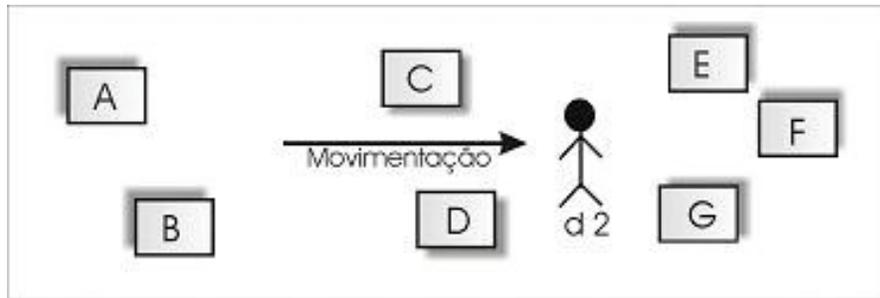


Figura 5.7 - Resultado da consulta submetida pelo usuário em nova posição da rede *ad hoc*

A posição da rede após o movimento é diferente da posição da rede na qual o usuário do dispositivo d_1 submeteu a consulta, assim como podem ser também as posições do usuário, conforme ilustrado na Figura 5.8. Esta figura ilustra a movimentação da rede e, além disso, ilustra os dispositivos d_1 e d_2 que tiveram suas posições alteradas.

O servidor móvel deve, então, gerenciar os resultados das consultas referentes a todos os usuários da rede *ad hoc*, considerando as mudanças de localização dos dispositivos e da rede de curta distância.

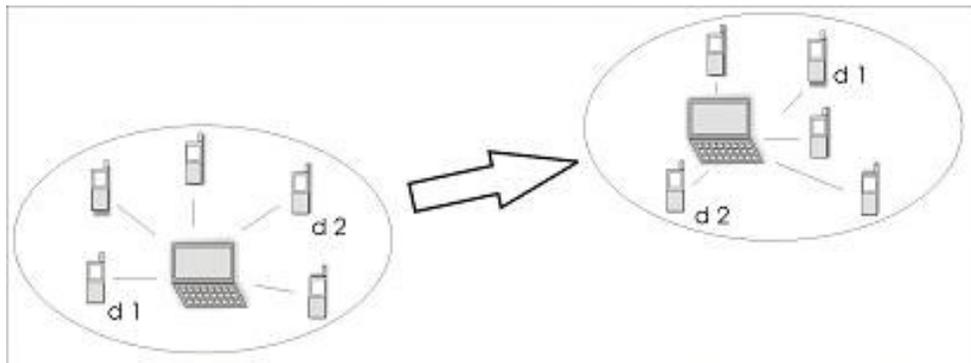


Figura 5.8 – Movimento da rede e mudança de posição dos dispositivos

Podem ser considerados exemplos típicos de utilização desse ambiente, grupos de turistas em visita a uma cidade, ou ainda, um grupo de gerência de tráfego (trânsito), que precisa se locomover para vários pontos na cidade.

5.4.2. Interação entre Clientes e Servidor Móvel

Um gerenciamento eficiente de cache deve manter em memória os itens que são mais prováveis de serem acessados, bem como manter a consistência de tais itens. Portanto, um gerenciador de cache deve realizar basicamente duas funções: determinar e executar políticas de substituição e promover atualização dos itens em cache. Nosso trabalho propõe um gerenciador que vai promover políticas de substituição e garantir a consistência dos itens em cache, como forma de melhorar a disponibilidade de dados e o processamento de CDL's em uma rede *ad hoc*.

Os mecanismos de gerência de cache tradicionais são normalmente baseados na hipótese de estabilidade de rede e uma razoavelmente alta largura de banda para transmissão. Esses dois requerimentos entram em conflito com as características de um ambiente móvel, no qual cada cliente móvel se conecta e desconecta com a rede freqüentemente, e a comunicação entre o cliente móvel e o servidor se faz via um canal de baixa largura de banda [45].

Por ser uma tarefa mais complexa que em ambientes distribuídos tradicionais, o gerenciamento de cache em ambientes móveis deve considerar o critério de localização, tanto dos dispositivos quanto dos itens fornecidos como respostas às consultas, nas estratégias de substituição e validação, bem como as limitações às quais os sistemas móveis sem fio estão submetidos, tais como *links* de comunicação de baixo desempenho, consumo de bateria e outras, conforme discutido no Capítulo 2.

Nosso servidor móvel é capaz de analisar uma consulta e verificar se esta pode ser respondida a partir dos dados que estão disponíveis localmente, pois ele mantém, como itens de cache, resultados de consultas prévias. Se isso for possível, a resposta à consulta é retornada de imediato ao cliente e, caso contrário, o servidor móvel encaminha solicitação ao servidor fixo. Caso a consulta possa ser respondida parcialmente, o servidor móvel solicita ao servidor fixo somente uma parte da resposta, que é definida após a pesquisa dos dados disponíveis em cache, diminuindo, desse modo, a quantidade de dados que trafegam pela interface aérea da rede de longa distância. Essa pesquisa é detalhada no Capítulo 6, onde seu algoritmo correspondente é apresentado e discutido.

Os dados que alimentam o espaço de memória reservado para cache no servidor móvel são enviados pelo servidor fixo como respostas às requisições que nele chegam. Na Figura 5.9 a seguir, pode ser observado que as requisições dos clientes móveis são atendidas através do servidor móvel, que

as responde a partir dos dados em cache ou as encaminha ao servidor fixo, que possui os recursos necessários para atender quaisquer solicitações de dados.

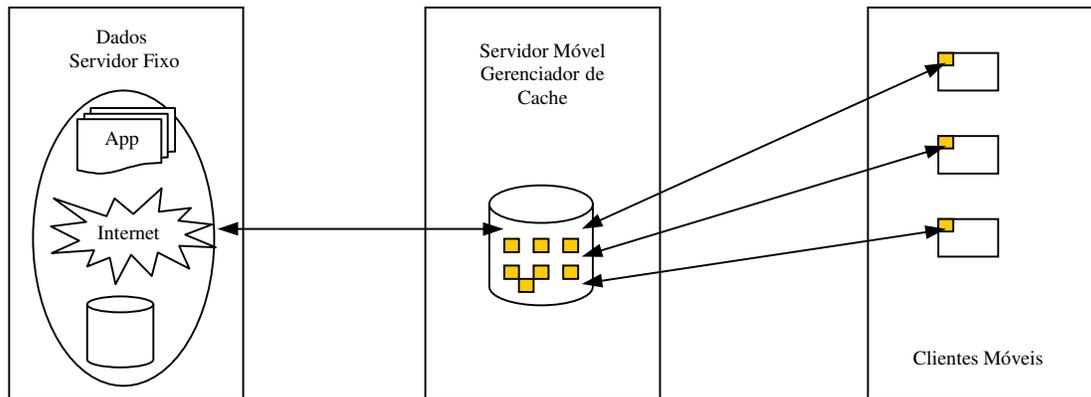


Figura 5.9 - Gerenciador de Memória do Servidor Móvel

Verificamos, portanto, que esses dados são armazenados nessa área de memória para prováveis acessos futuros, através das consultas submetidas pelos clientes ao servidor móvel na rede *ad hoc*.

5.4.3. Estratégias de Substituição

Por possuir tamanho limitado, a área de memória reservada para cache deve ser administrada a fim de que permaneçam nessa área os itens mais prováveis de serem acessados no futuro. O cache do servidor móvel armazena, então, resultados de consultas prévias como forma de melhorar o acesso a dados na rede *ad hoc* e diminuir os custos de comunicação. Dessa forma, ao receber uma requisição de dados que não estejam disponíveis em seu cache, o servidor móvel encaminha a solicitação ao servidor externo e, em seguida, de posse do resultado, armazena-o em memória cache.

Após algumas dessas operações de armazenamento, o espaço livre em memória torna-se insuficiente para que novos dados sejam salvos. Torna-se necessário, então, excluir itens para liberar o espaço necessário. Para realizar essas exclusões o servidor móvel deve levar em consideração a probabilidade de acesso dos dados, ou seja, os itens a serem escolhidos para sair do cache devem ser aqueles menos prováveis de serem requisitados no futuro.

O cálculo dessa probabilidade de acesso é feito levando em consideração, primeiramente, a localização do usuário. Por se tratar de consultas dependentes de localização, assumimos a proximidade de um item de dado como o critério de maior relevância na escolha dos objetos que devem permanecer em cache. Explicamos, neste capítulo, o mecanismo de substituição utilizado no gerenciador de cache em linhas gerais e apresentamos um e caso como exemplo para que sua funcionalidade seja melhor esclarecida. O algoritmo que responsável pela substituição é demonstrado formalmente no Capítulo 6, depois que forem introduzidas as estruturas de dados necessárias na implementação do protótipo do mecanismo de cache proposto.

A distância entre determinado item em cache (utilizado em resposta a uma consulta) e a posição atual da rede *ad hoc* é dada pela distância euclidiana, calculada a partir das coordenadas geográficas desses dois pontos. A distância euclidiana entre dois pontos é calculada através da seguinte fórmula:

$$\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \quad \text{(Fórmula 1)}$$

Onde x_1 e y_1 representam as coordenadas geográficas de um ponto P1 e x_2 e y_2 representam as coordenadas de um ponto P2, entre os quais se deseja calcular a distância. Portanto, a partir de agora, fica subentendido que, sendo necessário considerar a distância entre dois pontos, esta é calculada a partir da fórmula da distância euclidiana.

Se, no momento de inserir um determinado dado em memória, o servidor móvel verifica que é necessário liberar espaço, os itens escolhidos para serem eliminados são inicialmente aqueles que estiverem mais distantes da posição corrente do usuário.

O segundo critério utilizado na estratégia de substituição em cache do servidor móvel é a acessibilidade de determinado item em memória. Nesse sentido, se, no momento da escolha do item mais distante (que é o critério mais relevante de substituição), forem encontrados mais de um item igualmente distante do dispositivo móvel, a escolha dos dados a serem excluídos do cache do servidor móvel consistirá naqueles itens com menor número de acessos.

Consideremos o exemplo anteriormente apresentado na seção de CDL's (Seção 5.3.1). A ilustração do estado do cache após a consulta de d_1 (vide Figura 5.6) é apresentada na Figura 5.10

abaixo. O resultado retornado em resposta a essa consulta é armazenado nessa área de memória. A área mais escura da figura representa espaço em cache já preenchido com outros dados quaisquer.

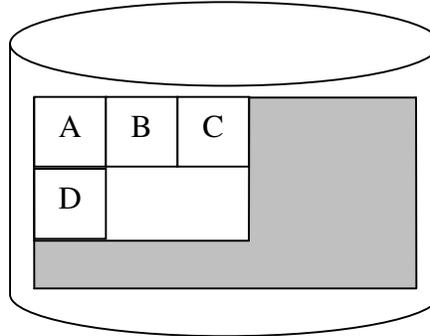


Figura 5.10 - Estado do cache após o armazenamento dos itens retornados como resposta à consulta submetida por D1

No momento em que o usuário submete a mesma consulta em nova posição (Vide Figura 5.7), o servidor móvel verifica se há dados em cache que satisfaçam tal consulta, através do procedimento de busca de itens XML (os algoritmos de busca de dados no cache é descrito e discutido no Capítulo 6). Após esse procedimento de busca, o gerenciador de cache do servidor móvel conclui que os restaurantes C e D, disponíveis em cache, são parte da resposta. O restante dos dados necessários para compor a resposta completa à consulta é, então, solicitada ao servidor fixo, que retorna os restaurantes E, F e G. Verificamos, através da figura, que não há espaço suficiente para guardar tais dados em cache. Sendo assim, é necessário que sejam realizadas substituições dos itens menos prováveis de serem acessados futuramente.

As distâncias entre os usuários e os restaurantes retornados como respostas às suas consultas são mostrados a seguir na Tabela 5.1.

Além das distâncias especificadas, a tabela contém a quantidade de acesso de cada um desses itens em cache, que será também utilizada na escolha dos dados a serem eliminados. Verifica-se que a quantidade de acessos dos dados retornados por último pelo servidor fixo (restaurantes E, F e G) na tabela é nula, visto que esses itens ainda não se encontram em cache no momento especificado.

Tabela 5.1 - Distância dos itens e quantidade de acessos dos dados em cache

Restaurantes	Distância de D1 no momento da requisição da primeira consulta (km)	Distância de D2 no momento da requisição da consulta em nova posição (km)	Quantidade de Acessos
A	4	8	2
B	4	8	4
C	3	1	3
D	2	1	2
E	7	3	-
F	8	3	-
G	7	2	-

De posse do resultado retornado pelo servidor fixo, o servidor móvel retorna a resposta ao cliente móvel e deve, em seguida, atualizar seu cache, efetuando o armazenamento dos dados referentes aos restaurantes E, F e G. Suponhamos que o tamanho livre em cache disponível para tal operação seja suficiente para armazenar somente dois desses itens. O gerenciador deve efetuar, portanto, uma eliminação, a fim de liberar espaço para armazenar o restante dos itens (vide Figura 5.11).

Para armazenar os dados referentes ao item G, o gerenciador do servidor móvel deve escolher o item menos provável de ser acessado para ser retirado do cache. O primeiro critério considerado é a distância entre os itens armazenados e a posição corrente do usuário. Considerando esse critério, o servidor móvel decide que, por estarem mais distantes do usuário que submeteu a consulta, os itens a serem eliminados serão A ou B. Como a distância da posição corrente do dispositivo é a mesma para ambos (vide Tabela 5.1), o gerenciador utiliza então o segundo critério, que é a quantidade de acessos. Dessa forma, como o restaurante B foi mais requisitado que o restaurante A, este último deve ser eliminado da área de memória reservada para cache, liberando, portanto, espaço para armazenar o novo item.

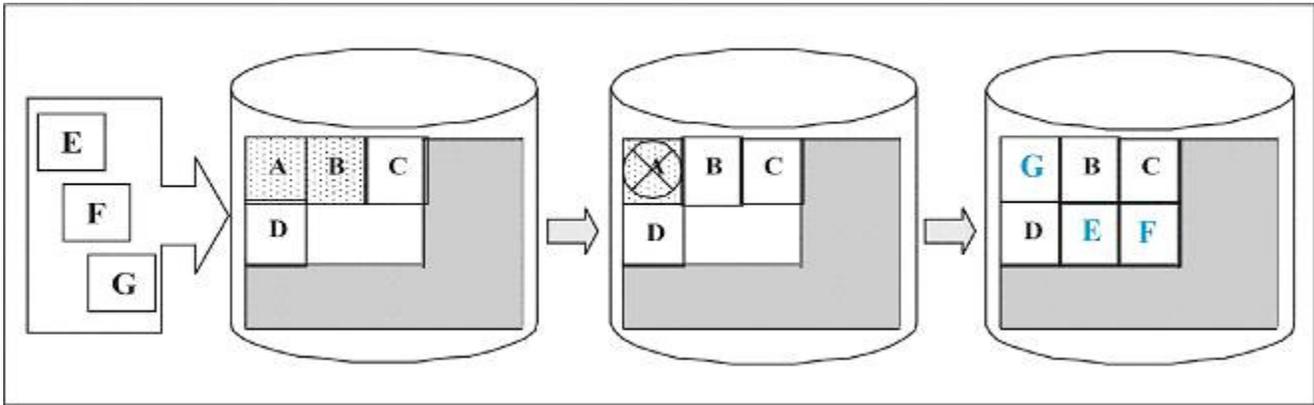


Figura 5.11 - Eliminação em cache para armazenar novos itens

5.4.4. Estratégias de Validação

5.4.4.1. Tipo de Validação Adotado

Como citado no Capítulo 2, os relatórios de validação imediatos não são adequados para validar dados em um ambiente móvel, pois geram aumento na utilização da interface aérea devido ao envio constante de informações de validação. Desse modo, no nosso trabalho, a estratégia escolhida para manutenção da coerência de dados no servidor móvel é baseada em um *método de validação por demanda* [71]. No método de validação por demanda, cada cliente é responsável por validar seus itens em cache quando necessário, ou seja, no momento em que determinado item for solicitado, o cliente verifica se este item pode ser utilizado ou se está obsoleto, baseando-se nas informações passadas pelo servidor.

Como estamos considerando a arquitetura de uma rede *ad hoc*, na qual um servidor móvel de dados oferece serviços aos clientes móveis, a estratégia de validação é então adaptada para que se torne mais adequada ao nosso trabalho. Dessa forma, o servidor móvel é responsável por manter a coerência dos itens em seu cache e de fornecer aos clientes da rede *ad hoc* informações confiáveis. Assim, ao receber uma solicitação de um determinado item, o servidor móvel deve verificar, caso este item esteja disponível localmente, se seu valor ali armazenado é válido, de acordo com a sua localização corrente, e, caso este valor esteja obsoleto, o servidor móvel deve solicitar ao servidor externo o valor atualizado

desse item, de forma a garantir que a resposta à consulta solicitada pelo usuário móvel seja composta de informações válidas.

Com o objetivo de validar os dados em cache, o servidor móvel pode enviar o identificador deste item ao servidor externo para checar sua validade. Assim, ao receber uma requisição de dados, o servidor externo decide, se estes itens são válidos, de acordo com a posição corrente do usuário e envia de volta o resultado da verificação. Esta verificação é feita somente no momento em que esse dado é requisitado, de maneira a evitar tráfego desnecessário de dados. Este é um método simples, porém o aumento do tráfego na rede de longa distância cresce consideravelmente devido a essas constantes verificações. Dessa forma, para melhorar o desempenho, são anexadas informações de escopo para os dados. O algoritmo que descreve com detalhes esse procedimento é apresentado no Capítulo 6 (Figura 6.3), depois que forem demonstradas as estruturas de dados utilizadas no gerenciador de cache e o conceito de escopo utilizado em nosso trabalho é descrito na seção a seguir.

5.4.4.2. Escopo

O escopo válido de um item é definido como o conjunto de áreas nas quais este item é válido [70]. O escopo de um item pode variar substancialmente dependendo de sua natureza. Por exemplo:

- Itens de dados que refletem serviços de área de larga abrangência, tais como informações climáticas, são associados a um escopo de várias células cobrindo dezenas ou até centenas de quilômetros;
- Condições de tráfego, que são informações com menor área de abrangência, são associadas a um escopo de tamanho de poucos quilômetros quadrados;
- Informações sobre disponibilidade de vagas em um estacionamento possuirão um escopo de poucas centenas de metros.

Dessa forma, a área geográfica referente ao escopo válido de um item depende da abrangência na qual tal informação é válida.

Utilizamos o conceito de escopo válido como forma de validar os dados no cache do servidor de dados móvel. Dessa forma, quando o servidor externo envia um determinado item ao servidor móvel, ele envia juntamente o seu escopo válido, isto é, envia o item e, além disso, a área na qual aquele valor pode ser considerado válido e, no momento em que este item for solicitado, o servidor

móvel analisa sua validade de acordo com o escopo que lhe é associado. Caso o valor do item em cache seja confiável, o servidor móvel o utiliza para compor a resposta à consulta. Caso contrário, ou seja, caso o valor desse item esteja obsoleto, o servidor móvel solicita ao servidor externo uma nova cópia de tal item.

5.4.4.3. Efeitos em Consultas de Somente Leitura

Em gerência de cache em ambientes distribuídos, deve-se levar em consideração a possibilidade de um determinado item ser modificado no servidor através de consultas de atualização (*update*). Desse modo, se o valor de um item de dado puder ser modificado no servidor por outros clientes, o cache do cliente provavelmente não poderá utilizar este valor em consultas subseqüentes. A verificação dessa possibilidade é feita para garantir a validade do valor de um determinado item retornado em resposta a uma consulta.

Em um ambiente móvel, clientes móveis usualmente podem aceitar um leve grau de dados desatualizados em troca de resposta rápida às consultas [45]. Além disso, a grande maioria das aplicações em um ambiente móvel gera mais operações de leitura que de escrita [26].

É necessário um controle para evitar que um dado desvie muito do seu atual valor. Como discutido na seção anterior, utilizamos o conceito de escopo válido para garantir uma certa margem de aceitação para dados dependentes de localização, que compõem o foco do nosso trabalho.

O tipo das consultas tratadas em nosso trabalho consiste em consultas somente de leitura, de modo que, fogem do nosso escopo inserções e modificações de dados pelos dispositivos móveis, bem como quaisquer operações de controle de concorrência, utilizadas para manter a consistência desses dados que estão sendo constantemente modificados.

Como utilizamos a validação por demanda, subentende-se que o servidor externo não notifica quando um determinado item for modificado. Dessa forma, como a validação de um item é feita somente quando este item é solicitado, é possível que o cache mantenha dados muito obsoletos e inválidos em memória. Logo, para evitar que um item permaneça um longo período de tempo em cache, estimamos também um “tempo-limite” para cada item.

Desse modo, cada item em cache possui uma marca de tempo, na qual este item é descartado quando o tempo a ele associado expira. Quando essa marca de tempo é atingida, o item é eliminado do cache, de modo a liberar espaço dessa área de memória. O intervalo de tempo no qual um item expira

deve ser definido de acordo com a utilização do cache e o fluxo de requisições de consultas na rede considerada. Assim, se há uma grande quantidade de requisições em uma rede *ad hoc*, num determinado período de tempo, o “tempo limite” dos itens mantidos em cache nessa rede deverá ser menor do que o tempo limite dos itens de uma rede de pouca utilização, pois a área de memória reservada para cache é mais concorrida no primeiro caso.

Esse tempo limite estabelece o tempo máximo que um item pode ser considerado válido no escopo a ele associado. Essa abordagem visa evitar que sejam utilizados dados muito antigos armazenados no cache do servidor móvel de dados.

A rotina para controlar o tempo limite de cada item em cache é descrita na Figura 5.12 a seguir.

```

Rotina ControlaTempoLiquido{
    While (! percorrer todos os itens em cache){
        Selecciona elemento E;
        If (tempoEmCache(E) > tempoLimite){
            Elimina E do cache;
        }
    }
}

```

Figura 5.12 - Rotina para controle de tempo limite do cache

Essa rotina deve ser executada em intervalos de tempos pré-estabelecidos, dependendo do fluxo de requisições e da taxa de movimentação da rede considerada. Essa taxa de movimentação é calculada pelo servidor móvel que decide, então, sobre o valor do tempo limite e o intervalo no qual a rotina de controle desse tempo líquido será executada.

5.5. Manutenção dos Dados em Memória

O crescente uso de XML como linguagem padrão para troca de dados em sistemas distribuídos nos motivou a efetuar a manutenção dos dados no cache do servidor móvel no formato XML. Essa escolha traz a vantagem de se trabalhar com um formato único para troca e manipulação de dados, ilustrado na Figura 5.13.

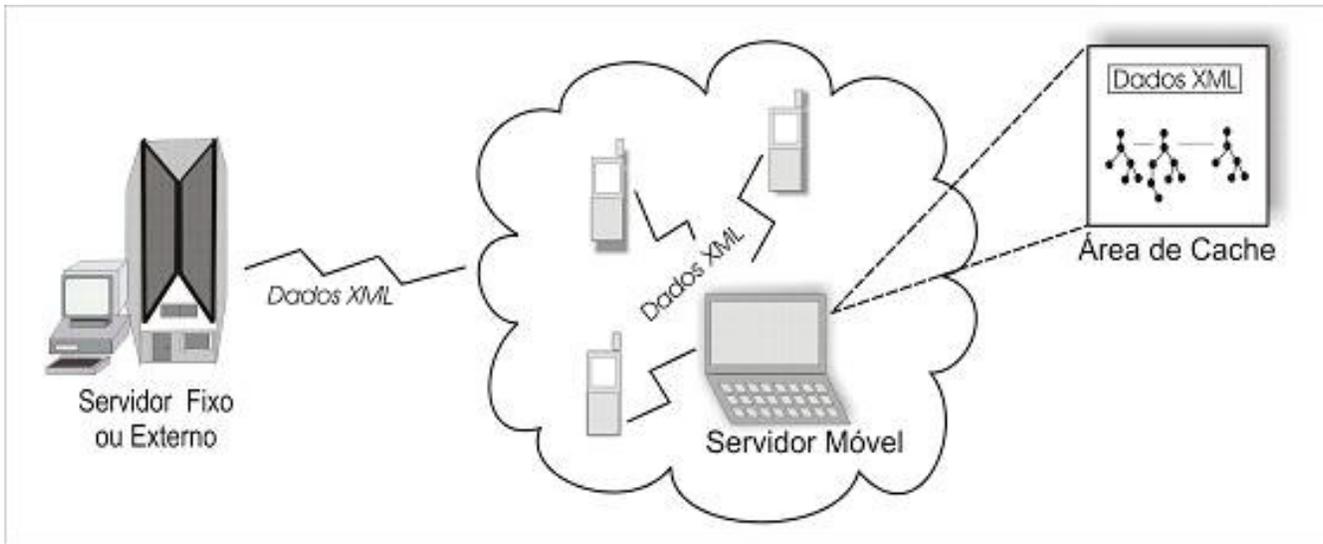


Figura 5.13 - Manutenção e transporte de dados no formato XML

Vários trabalhos [44],[55],[56] já mostraram que *caching* de páginas não é a melhor opção para ambientes móveis sem fio, visto que o custo de comunicação é significativo, já que são enviados dados desnecessários em uma página requisitada. A Tabela 5.2 apresenta um quadro-resumo, que ilustra as desvantagens encontradas em armazenamento de páginas em cache.

Tabela 5.2. Quadro de desvantagens de cache de páginas.(Adaptado de [56])

Item	Cache de Páginas	Razão
Custo de comunicação	Alto	Dados desnecessários são transferidos ao dispositivo que solicitou a consulta
<i>Overhead</i> do espaço em cache	Alto	Espaço é preenchido com dados que não foram solicitados em consultas anteriores
Paralelismo no processamento de consultas	Difícil	Difícil determinar exatamente quais os dados que faltam para compor determinada informação
Utilização em computação móvel	Ineficiente	Não há praticamente autonomia para usuários móveis em caso de desconexão.

Em virtude desse fato, propomos como unidade de armazenamento em cache os próprios objetos retornados como resposta às consultas pelo servidor externo. Esses dados são organizados no formato lógico de XML (estrutura de árvore DOM), no qual o acesso a tais informações ocorre através de suas expressões de caminho correspondentes.

5.6. Conclusão

Neste capítulo, apresentamos as características de funcionamento do mecanismo gerenciador de cache proposto neste trabalho, bem como as estratégias utilizadas para que esse gerenciador seja adaptado, eficientemente, para o ambiente móvel sem fio.

De forma resumida, é possível observar que, apesar de existirem vários trabalhos na literatura referentes ao gerenciamento de cache em ambientes móveis [8][21][34][41][44][45], o mecanismo proposto neste trabalho possui características que o tornam mais eficiente para o ambiente considerado. Dentre essas características, podemos citar a atuação em um servidor de dados móvel, aumentando, assim, a capacidade do cache e tornando disponível dados comuns aos usuários da rede *ad hoc* e o tratamento de dados no formato XML, mantendo, desse modo, um formato único para troca e manipulação de dados.

No capítulo seguinte, aprofundamos nossa discussão sobre o mecanismo de gerência de cache proposto, apresentando o programa computacional implementado para analisar a eficiência dos métodos propostos e os resultados obtidos a partir dos experimentos realizados.

CAPÍTULO 6

IMPLEMENTAÇÃO DE PROTÓTIPO E EXPERIMENTOS

Neste capítulo são apresentadas as características de implementação apresentadas no desenvolvimento do programa computacional utilizado para simular, de forma simplificada, o gerenciador de cache que atua em um servidor de dados móvel, proposto neste trabalho. Na Seção 6.1 são apresentadas as configurações gerais do programa, na Seção 6.2 é introduzida uma visão geral de funcionamento, a Seção 6.3 apresenta as estratégias e algoritmos propostos no trabalho. Na Seção 6.4 são descritas as estruturas de manipulação utilizadas na implementação, na Seção 6.5 é ilustrado, de forma detalhada, o funcionamento do programa e, finalmente, na Seção 6.6 são apresentados estudos comparativos.

6.1. Configurações do Protótipo do Gerenciador de Cache

O programa criado para simular o gerenciador de cache apresentado neste trabalho foi construído utilizando a linguagem C++ no ambiente de desenvolvimento *Borland C++ Builder*.

Os documentos XML, provenientes de respostas a consultas, são mantidos em cache e, no momento que alguma consulta é submetida, esses documentos são analisados com o intuito de descobrir se os dados neles presentes são úteis para compor a resposta para tal consulta. Para percorrer esses documentos XML mantidos em cache, utilizamos como *parser* o Xerces [1]. O Xerces é um *parser* de validação XML, escrito em C++, que auxilia uma aplicação na capacidade de ler e escrever dados XML [1]. Para isso, uma biblioteca de Xerces compartilhada está disponível para geração, manipulação e validação de documentos XML. Dessa forma, a utilização do Xerces auxilia o gerenciador de cache proposto na identificação de dados que serão parte da resposta a uma determinada consulta. Desse modo, percorrendo árvores DOM, geradas com a ajuda desse *parser*, é possível

identificar dados úteis mantidos em cache de forma mais eficiente, visto que a estrutura de árvore é o meio padrão para manipulação de itens XML.

6.2. Visão Geral do Funcionamento do Gerenciador de Cache

Descrevemos, nas seções seguintes, um programa simples utilizado para calcular o percentual de reaproveitamento de dados, quando utilizando o gerenciador de cache proposto. Esta implementação foi realizada com o intuito de verificar se há, realmente, uma diminuição do fluxo de dados na comunicação com um servidor fixo da rede de longa distância, através da utilização de um gerenciador de cache disponível em um servidor móvel. Neste sentido, pretendemos mostrar, com esse programa, que o mecanismo proposto aumenta a disponibilidade de dados na rede *ad hoc*, diminuindo, conseqüentemente, o custo geral referentes à submissão de CDLs.

Os itens XML em cache possuem atributos de localização, representados por coordenadas geográficas, pois, como introduzido no Capítulo 5, o intuito do gerenciador é reaproveitar dados referentes a respostas de CDLs anteriores.

É utilizado um diretório que representa o cache do servidor móvel, onde ficam mantidos fragmentos de documentos XML, os quais contêm respostas de consultas anteriores submetidas pelos clientes móveis. O programa possui duas interfaces de atuação: uma correspondente à interação entre o gerenciador de cache do servidor móvel e clientes móveis, e outra que corresponde à interação entre o gerenciador de cache do servidor móvel e o servidor fixo. O fluxo geral dessas interações é apresentado no diagrama de seqüência, na Figura 6.1 a seguir.

Na primeira interação (entre servidor móvel e clientes móveis), a entrada para o programa consiste de um arquivo de requisições, no qual estão listadas expressões de caminho que correspondem à consulta de um usuário. Neste momento, o gerenciador verifica a existência desses itens no cache local, através do *parsing* dos documentos ali mantidos, identificando a parte da consulta que pode ser respondida localmente.

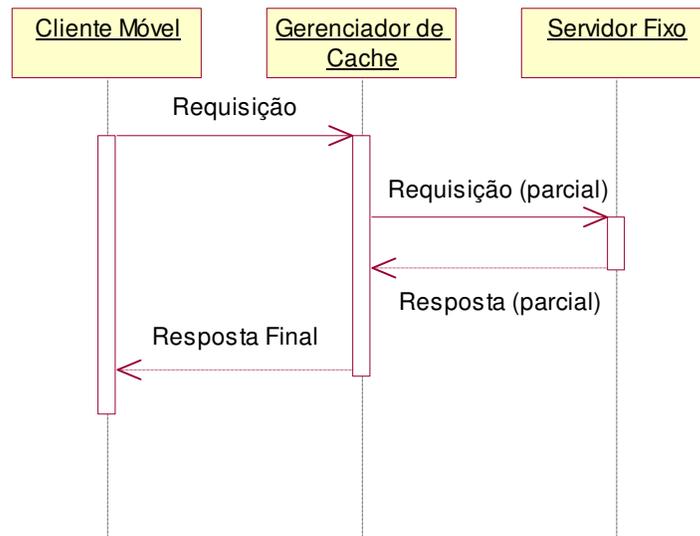


Figura 6.1 - Diagrama de Seqüência de Interação entre entidades

Discutimos no Capítulo 5 que a busca dos itens disponíveis localmente é feita em dois níveis, sendo que, o primeiro consiste na memória do servidor móvel e o segundo, em seu disco. Logo, é importante ressaltar que, nessa implementação, consideramos apenas o primeiro nível, pois a busca de itens no disco acarretaria tratamento para os dados ali replicados, que está além do escopo deste trabalho, sendo, porém, uma importante funcionalidade a ser desenvolvida futuramente. Dessa forma, no programa desenvolvido, os itens contidos no arquivo de requisições que não estiverem disponíveis em cache, são solicitados, então, ao servidor externo.

O servidor fixo e o servidor móvel utilizam o FoX como sistema de banco de dados. Desse modo, assumimos que os dois servidores trocam informações e fazem o tratamento de seus dados da mesma maneira (através de itens de dados XML), o que implica, portanto, que ambos os servidores processam as requisições e retornam os dados de maneira compatível entre eles.

Nesse sentido, na segunda interação (entre servidor móvel e servidor fixo), a entrada para o programa passa a ser um arquivo XML contendo o resultado das requisições encaminhadas ao servidor fixo. Neste momento, é analisado se o espaço é suficiente para guardar tal documento e, se necessário, é realizada substituição de dados para armazenar em cache as novas informações.

Uma extensão do diagrama de atividade acima apresentado é mostrada, a seguir, na Figura 6.2, como forma de ilustrar as duas interações consideradas neste trabalho. Além disso, são descritos na figura os processos que o mecanismo gerenciador de cache deve gerar após o recebimento de dados dos clientes móveis e servidor fixo em cada interação.

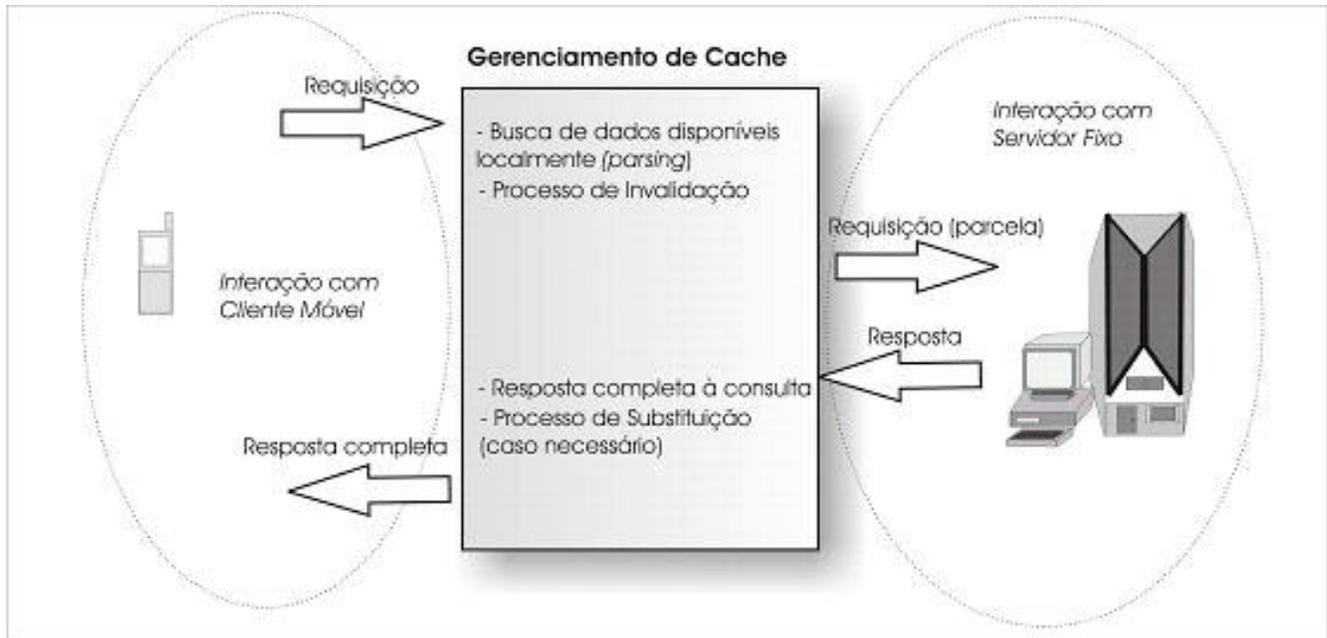


Figura 6.2 - Interação entre Clientes Móveis, Gerenciador de Cache e Servidor Fixo

As entradas e saídas do programa são feitas através da utilização de arquivos (.dat). Os motivos dessa escolha e os meios de manipulação dessas estruturas são apresentados de forma mais abrangente na Seção 6.4.

Além disso, é importante ressaltar que o programa não realiza nenhum tipo de processamento de consulta, pois este deve ser realizado em um módulo superior ao gerenciamento de cache, como ilustrado no Capítulo 5.

Conforme já citado, são mantidos em cache resultados de consultas prévias, os quais ficam guardados em documentos XML. Desse modo, os documentos XML funcionam como uma espécie de *cluster* no gerenciador de cache, onde itens relacionados de requisições já processadas são agrupados no mesmo documento e são tratados como um todo no momento de substituição em cache, quando se

fizer necessário. Esse procedimento torna mais viáveis tanto procedimentos de substituição, quanto funções de busca de itens de dados em cache.

6.3. Estratégias e Algoritmos

6.3.1. Interação com Clientes Móveis

O servidor móvel recebe requisições dos clientes móveis da rede *ad hoc*, que são fornecidas ao programa através de um arquivo contendo as expressões de caminho que representam essas consultas. A partir desse arquivo de requisições fornecidas como entrada, o programa analisa se existe parte da resposta disponível localmente utilizando, para isso, o algoritmo de busca descrito a seguir, na Figura 6.3.

Algoritmo BUSCA_CACHE

Entrada: £ - Lista de expressões de caminho referentes a consultas de usuários

Saída: θ - Estrutura que contém o conjunto de itens a serem retornados ao cliente, como resposta à consulta

£ atualizada, contendo somente as requisições que não estão disponíveis em cache

//-- Procedimento que retorna verdadeiro se o escopo de determinado item for válido e
 //-- falso, caso contrário.

Booleano EscopoValido(x,y)

```
{
    x_atual = coordenadaX da posição central da rede ad hoc;
    y_atual = coordenadaY da posição central da rede ad hoc;
    dist_euclidiana = sqrt ((x_atual - x)2 + (y_atual - y)2);
    if dist_euclidiana <= distanciaLimite then
        return true;
    else
        return false;
}
```

//-- Procedimento de busca, nas estruturas em cache

```

 $\theta \leftarrow \{\};$ 
 $\mathcal{L} \leftarrow$  lista de requisições;
while (! percorre todas as estruturas DOM presentes em Cache)
{
  Seleciona elemento E;
  If ( $E \in \mathcal{L}$ )
  {
    x = coordenadaX de E;
    y = coordenadaY de E;
    If (EscopoValido(x,y))
    {
      //- Elemento serve para ser retornado para o cliente
      Retira (E,  $\mathcal{L}$ );
      Insere (E,  $\theta$ );
    }
  }
}

Envia  $\mathcal{L}$  ao servidor externo;
Adiciona o resultado recebido da requisição de  $\mathcal{L}$  à  $\theta$ ;
Envia  $\theta$  ao cliente;

```

Figura 6.3 – Algoritmo de busca em cache

Após a execução deste algoritmo, a lista de expressões de caminho \mathcal{L} , fornecida como entrada, é atualizada, de modo que constarão nela somente as requisições dos objetos que não se encontram disponíveis no cache local. Além disso, a validade dos dados é garantida nesse momento, através da verificação do escopo, que é calculado a partir da distância euclidiana, conforme pode ser observado no algoritmo. Desse modo, o algoritmo de validação está contido no algoritmo de busca.

Analisando o algoritmo acima apresentado, podemos verificar que é no momento de busca de itens que é realizado o *parsing* dos itens em cache, com a intenção de percorrer todas as informações contidas em cache para que seja recuperada, localmente, parte da consulta submetida pelo usuário.

Por ocorrer o *parsing* dos dados em cache e por existirem estruturas de repetição aninhadas, a quantidade de passos necessários para essa funcionalidade apresenta-se razoavelmente elevada, aumentando, dessa forma, a complexidade do algoritmo. Entretanto, para o propósito especificado, essa busca compensa tal esforço computacional, visto que esse procedimento visa diminuir o fluxo de dados na interface aérea da rede celular, que corresponde ao maior custo de comunicação na arquitetura considerada. Além disso, a quantidade de *clusters* de dados em cache não pode ser muito grande, pois a

área de memória reservada para cache é limitada, o que evita, conseqüentemente, *overhead* de processamento referente a *parsing* de dados.

6.3.2. Interação com Servidor Fixo

Na interação com o servidor fixo, ao receber as respostas das requisições, o servidor móvel deve armazenar este resultado em cache, eliminando os itens menos prováveis de acesso futuro, caso o espaço em cache esteja completamente ocupado. Na Figura 6.4 a seguir, é apresentado o algoritmo responsável pela implementação das estratégias de substituição do gerenciador de cache, a serem utilizadas quando se fizer necessário.

Algoritmo SUBSTITUICAO

Entrada: Δ – Novo item retornado pelo servidor fixo, como resultado à uma requisição.

Saída: Estado do cache atualizado com novas informações

Recebe novo documento Δ ;

If (Total_Itens_Cache == Limite_Cache)

{

 /-- É necessário efetuar substituição

 Calcula distancia_euclidiana para cada *cluster*;

 Array Vitimas[] = item(ns) de maior distancia_euclidiana

 If (size(Vitimas) > 1)

 {

 Item = procura item menos acessado;

 Elimina Item do cache;

 }

 Else

 Elimina (Vitima[0]) do cache;

 }

Armazena Δ em cache;

Figura 6.4 – Algoritmo de substituição em cache

Conforme discutido em seções anteriores e como pode ser observado no algoritmo acima apresentado, os itens em cache são agrupados em documentos XML, os quais são tratados como um todo na escolha dos segmentos a serem eliminados do cache. Cada agrupamento possui, então, uma referência de localização (em coordenadas geográficas) que fornecem uma medida aproximativa da posição geral dos itens que ali estão e um número correspondente ao total de acessos daquele *cluster*.

Analisando o algoritmo de substituição, podemos observar que, no pior caso, ele pode percorrer todos os *clusters* dispostos em cache duas vezes, se, por acaso, todos estes elementos possuírem a mesma distância euclidiana. Assim, após calcular as distâncias referentes a todos os dados em cache, o algoritmo deve percorrer novamente todos esses dados, com o intuito de selecionar o item menos acessado.

Entretanto, assim como o primeiro algoritmo, esse aumento no número de passos no processo de substituição compensa pelo propósito de manter os dados mais prováveis de acesso futuro disponíveis em cache e, conseqüentemente, pela diminuição do fluxo de dados na interface aérea da rede de longa distância.

É importante observar que, no pior caso, o mecanismo de gerenciamento de cache proposto atua conforme um critério já utilizado em sistemas tradicionais (considerando o critério de quantidade de acessos), não invalidando, portanto, o tratamento de outros tipos de consultas (além das CDL's). Entretanto, é necessário adaptar o gerenciador de cache para que este possa atuar tanto com as estratégias aqui apresentadas quanto com estratégias tradicionais de gerenciamento de cache, como forma de atuar independentemente do tipo de consulta. Tal adaptação, porém, fica como ponto a ser discutido e apresentado em um trabalho futuro.

6.4. Estruturas de Manipulação

Nesta seção, descrevemos o modo pelo qual os dados são manipulados no programa de forma a esclarecer como persistem e como são tratadas as informações referentes ao gerenciamento de cache implementado.

6.4.1. Arquivos

Como citado na seção 6.1, as entradas e algumas saídas do programa são feitas no formato de arquivo (.dat), tanto na interação com o servidor fixo quanto na interação com os clientes móveis. Estes arquivos representam a manipulação e a troca de dados entre clientes móveis, servidor fixo e servidor móvel, simulando informações necessárias ao gerenciador de cache em um ambiente móvel sem fio real.

Nesse sentido, apresentamos, a seguir, uma tabela onde, para cada arquivo, são associados seu tipo e sua representação no programa, com o intuito de esclarecer o papel de cada estrutura de arquivo utilizada na implementação.

Tabela 6.1 – Arquivos utilizados na execução do programa

Arquivo	Tipo	Papel
Cache.dat	Entrada	Representa os documentos mantidos em cache em um dado momento.
Consultas.dat	Entrada	Lista de requisições (em XPath) que representa a consulta do usuário depois de passar pelo processador de consultas.
Respostas.dat	Saída	Lista de itens de dados em cache representando parte da resposta à consulta de entrada, itens esses disponíveis no cache local.
Exporta.dat	Saída	Lista de requisições (em XPath) que indica a solicitação que será encaminhada ao servidor fixo.

Além desses arquivos, outros foram criados para auxiliar na persistência dos dados referentes ao gerenciador de cache (por exemplo, a quantidade de acessos dos itens), utilizadas nas estratégias de validação e substituição. Esses arquivos foram utilizados devido à dificuldade em associar a um documento XML em cache características necessárias ao gerenciador (como quantidade de acessos e coordenadas de localização de determinado *cluster*). Tal dificuldade foi acarretada pelo fato de trabalharmos com um *parser* já implementado, o que torna inviável, portanto, o acesso para alteração de estado de determinado item no Xerces.

6.4.2. Árvores DOM

Como introduzido anteriormente, a procura de itens que serão utilizados para compor a resposta de uma consulta é feita através de uma verificação nos documentos XML em cache. Já sabemos que estruturas de árvores DOM facilitam manipulação de dados XML e, por esse motivo, no momento de busca de itens solicitados em consulta, utilizamos o Xerces para montar as árvores DOM referentes a esses documentos e, a partir daí, realizar a busca e capturar um determinado item, caso ele esteja na relação de requisições e, além disso, que esteja dentro do escopo válido.

A Figura 6.5. a seguir ilustra o cenário depois de realizado o *parsing* e o estado do cache, no qual procedimentos de busca e de captura são realizados.

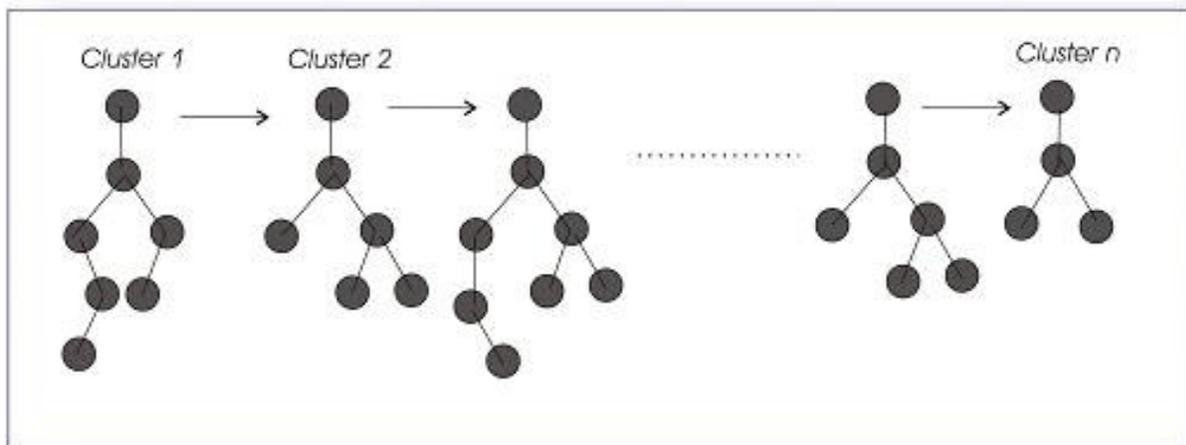


Figura 6.5 - Representação de documentos em Cache

6.5. Funcionamento

Nesta seção apresentamos o funcionamento do programa como forma de complementar as informações descritas nas seções anteriores e de ilustrar a maneira pela qual foram coletados os dados para a elaboração dos estudos comparativos, que serão apresentados na seção seguinte.

A utilização do programa acontece via console, onde é feita a chamada ao gerenciador, passando como parâmetro o dado de entrada necessário para cada interação.

Para iniciar, então, a utilização do programa, reproduzimos a interação entre servidor móvel e clientes móveis, chamando o programa Gerenciador para pesquisar os dados mantidos em cache nesse momento, fornecendo como parâmetro o arquivo no qual estão as expressões de caminho referentes à requisição do usuário. Na primeira utilização, percebe-se que nada existe na área de memória reservada para cache, sendo necessário, portanto, encaminhar a solicitação completa (descrita em Consulta.dat) para o servidor fixo. A Figura 6.6 ilustra tal procedimento. Os totais de itens em cache, o número de requisições (em expressões de caminho) disponíveis localmente para uso e o total dessas requisições que são encaminhadas ao servidor fixo são apresentadas na tela, para que se tenha uma idéia de um percentual de disponibilidade de dados.

```

Prompt de Comando

C:\Cache\Gerenciador Consulta.dat
*****
Pesquisando dados em Cache...
*****
Total de itens em cache: 0
Total de itens encontrados localmente: 0
Total de itens encaminhados ao servidor fixo: 15

C:\cache\Gerenciador Cache.dat_

```

Figura 6.6 – Funcionamento do programa simulando interação com clientes móveis

Após algumas submissões de consultas, o cache do servidor móvel vai sendo povoado, e alguns dados presentes nas requisições seguintes passam a ser encontrados localmente, diminuindo, dessa forma, as requisições ao servidor fixo.

Como, para cada requisição de dados enviada ao servidor fixo, há um documento XML que corresponde a uma resposta para essa requisição, em algum momento, o limite de armazenamento do cache será atingido, o que força o procedimento de eliminação dos itens menos prováveis de acesso futuro.

Na Figura 6.7 seguir é apresentada a interação com o servidor fixo, no qual o gerenciador recebe um documento XML e deve armazená-lo, seguindo os passos descritos no algoritmo de substituição apresentado na seção anterior. Nessa chamada ao programa, deve ser apresentado o parâmetro -S que indica que a interação é com o servidor fixo, passando, além disso, o nome do documento XML que deve ser inserido no cache.

```
Prompt de Comando

C:\Cache\Gerenciador -S restaurantes.xml
*****
Resposta OK.
Verificando espaço em cache...
Efetuando substituição...
Item Eliminado: hoteis.xml

C:\cache\Gerenciador Cache.dat_
```

Figura 6.7 – Funcionamento do programa simulando interação com servidor fixo

6.6. Estudos Comparativos

Com o intuito de mostrar que o gerenciamento de cache proposto melhora a disponibilidade de dados dependentes de localização em uma rede *ad hoc*, foram realizados estudos e comparações, utilizando amostras de CDLs e seus respectivos documentos XML de resposta. Não foi realizada nenhuma simulação em um ambiente móvel real, pois para o intuito desejado, o armazenamento em cache de dados referentes a respostas de CDLs é suficiente para alcançarmos o propósito de calcular a taxa de reaproveitamento desse tipo de informação.

6.6.1. Parâmetros de Utilização

Para a realização dos testes, foi desenvolvido um programa semelhante ao implementado para o gerenciador de cache do servidor móvel, o qual também realiza procedimentos de busca, substituição e validação de cache. Entretanto, este programa possui a característica de que as estratégias utilizadas no seu gerenciamento de cache são baseadas apenas na quantidade de acessos dos itens ali mantidos, de modo que as propriedades de mudança de localização dos dispositivos móveis são ignoradas.

As estratégias de validação implementadas neste segundo programa são idênticas às estratégias de validação implementadas no gerenciamento de cache para o servidor móvel, pois a restrição referente às CDL's (onde o critério de localização é explicitamente solicitado) implica que a validade das informações contidas no cache do servidor móvel é garantida conforme a posição dos itens em relação à posição corrente do dispositivo que realizou a consulta. Neste sentido, os itens que são retornados como resposta a uma determinada requisição devem ser validados conforme um raio de escopo que lhe é associado, fato este que torna sem sentido uma comparação com quaisquer estratégias que não considerem o critério de localização. Além disso, conforme já citado no Capítulo 5, a garantia de coerência de cache baseada em atualização de dados, foge do escopo deste trabalho inicial, de modo que comparações entre estratégias de validação ficam como um tópico para trabalhos a serem realizados futuramente.

Dessa forma, realizamos comparações entre um método que implementa suas estratégias de substituição, levando em consideração somente a quantidade de acessos dos itens em cache (critério este utilizado em técnicas tradicionais de gerenciamento de cache) e um método de substituição que considera primeiramente a localidade dos dados e, em segundo lugar, a quantidade de acessos que lhe são associadas. O último método representa o mecanismo de gerência de cache para servidores móvel de redes *ad hoc* proposto neste trabalho, conforme discutido no Capítulo 5 e apresentado, de maneira resumida, no algoritmo ilustrado na Figura 6.4.

As requisições que representam as consultas dos usuários foram escolhidas randomicamente, dentre todas as expressões de caminho possíveis geradas a partir de uma base de documentos XML escolhida para a realização dos testes. Essas requisições consistem de um arquivo composto por dez expressões de caminho, onde cada uma dessas expressões corresponde a um item de dado que está sendo solicitado em determinado momento.

A localização da rede, também gerada randomicamente no início dos testes, é modificada depois de 20 requisições com o intuito de analisar como ambos os métodos se comportam diante da mudança de localização dos dispositivos, já que esta é uma característica essencial nos sistemas móveis sem fio. A representação em um ambiente móvel sem fio real deste procedimento é ilustrada na Figura 6.8 a seguir.

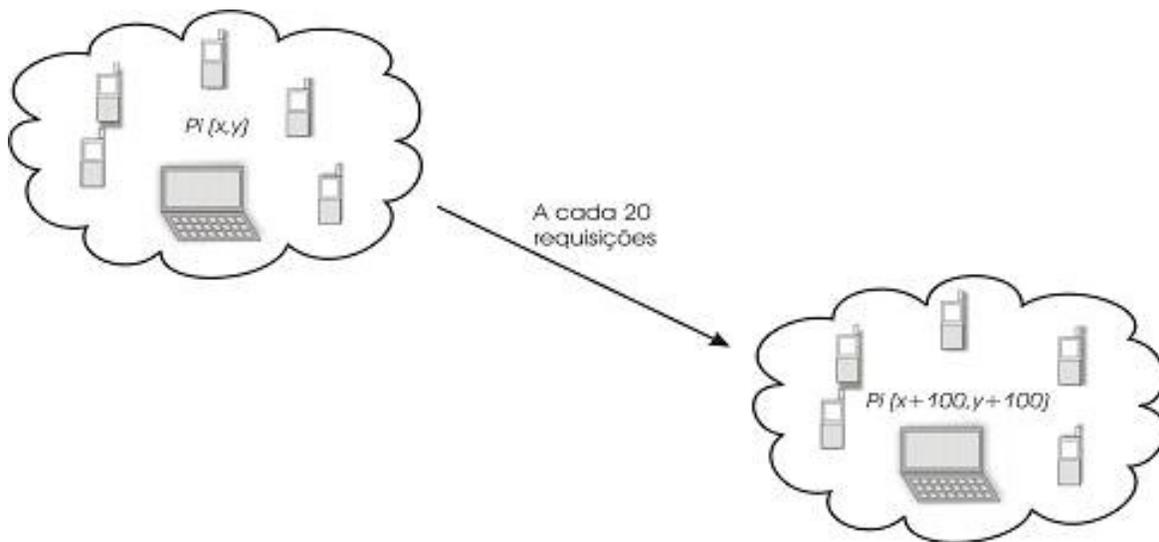


Figura 6.8 – Mudança de localização depois 20 requisições

A área de cache é iniciada com 60% de sua capacidade, sendo que a cada consulta, um novo fragmento de documento XML, referente à parte da resposta da solicitação que foi fornecida pelo servidor externo, deve ser inserido em cache.

Alguns parâmetros foram fixados com o intuito de viabilizar os testes e podem ser conferidos, de forma resumida, a seguir:

- Tamanho máximo de cada *cluster*: 18KB;
- Limite do Cache: 10 *clusters*;
- Cache inicia com 60% de sua capacidade;
- Cada item é considerado válido, de acordo com o escopo que lhe é associado;
- A quantidade de acesso dos itens é atualizada a cada requisição de dados;

- Depois de 20 requisições, uma nova localidade é atribuída à posição central da rede *ad hoc*;
- Um arquivo contendo o registro dos 10 últimos elementos descartados é mantido para efeito de histórico de substituição.

Em relação ao último tópico acima mencionado, esse histórico foi mantido para viabilizar comparações entre as escolhas dos itens eliminados nos dois métodos considerados nos testes, tornando possível, desse modo, verificar se as últimas eliminações dos elementos em cache foram as mais adequadas naquele momento, para cada estratégia de substituição considerada. A composição do arquivo que mantém tal histórico é ilustrada na Figura 6.9 a seguir.

Depois que tal registro de itens descartados estiver razoavelmente povoado (contendo um mínimo de 3 itens) são realizados procedimentos de busca nos itens ali indicados, de maneira que, para cada nova requisição de dados, é realizada uma pesquisa, para localizar os objetos solicitados, tanto nos itens mantidos em cache quanto nos itens recentemente descartados. Desse modo, é possível induzir que, quanto maior o percentual de itens encontrados nos elementos recentemente descartados, maior a probabilidade de que os métodos de substituição utilizados em determinado momento não sejam adequados para o ambiente em questão.

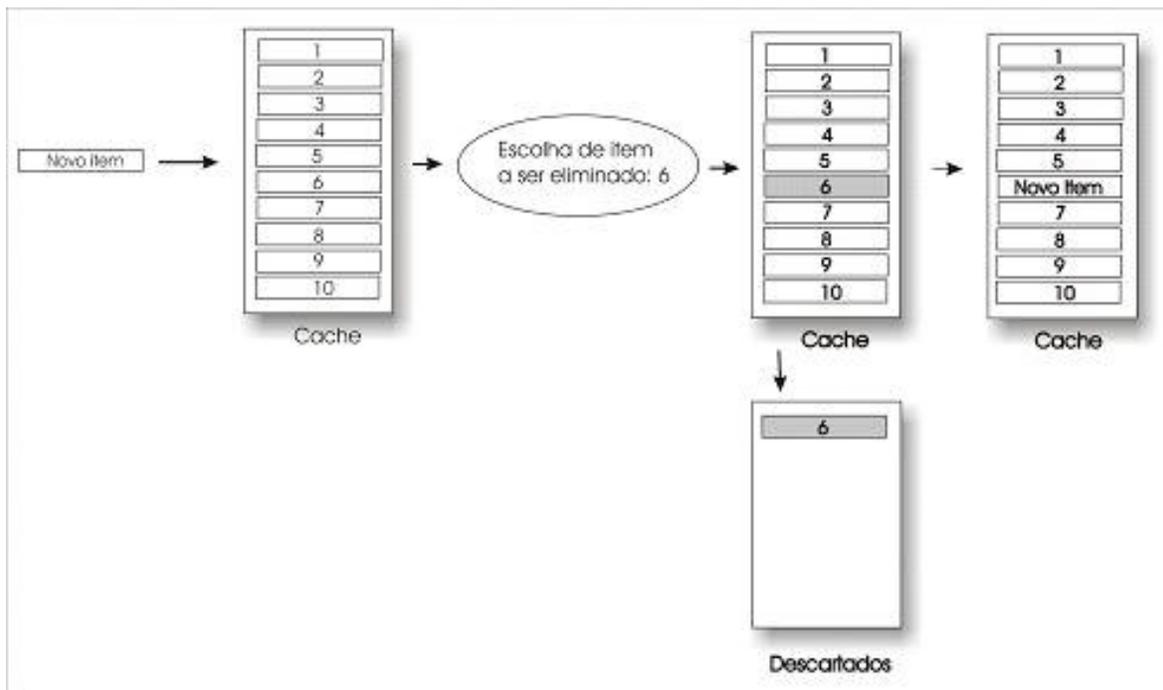


Figura 6.9 - Manutenção de registro dos itens recentemente descartados

6.6.2. Resultados Obtidos

Conforme introduzido na seção anterior, os testes realizados são baseados no percentual de aproveitamento das informações em cache diante de determinada requisição do usuário e, além disso, nos dados recentemente descartados.

O gráfico apresentado na Figura 6.10 ilustra o comportamento dos métodos observado em 80 requisições conforme os parâmetros introduzidos na seção anterior. O eixo das coordenadas corresponde ao percentual de itens solicitados e encontrados localmente em cache, enquanto o eixo das abscissas corresponde ao fluxo das requisições realizadas nos testes.

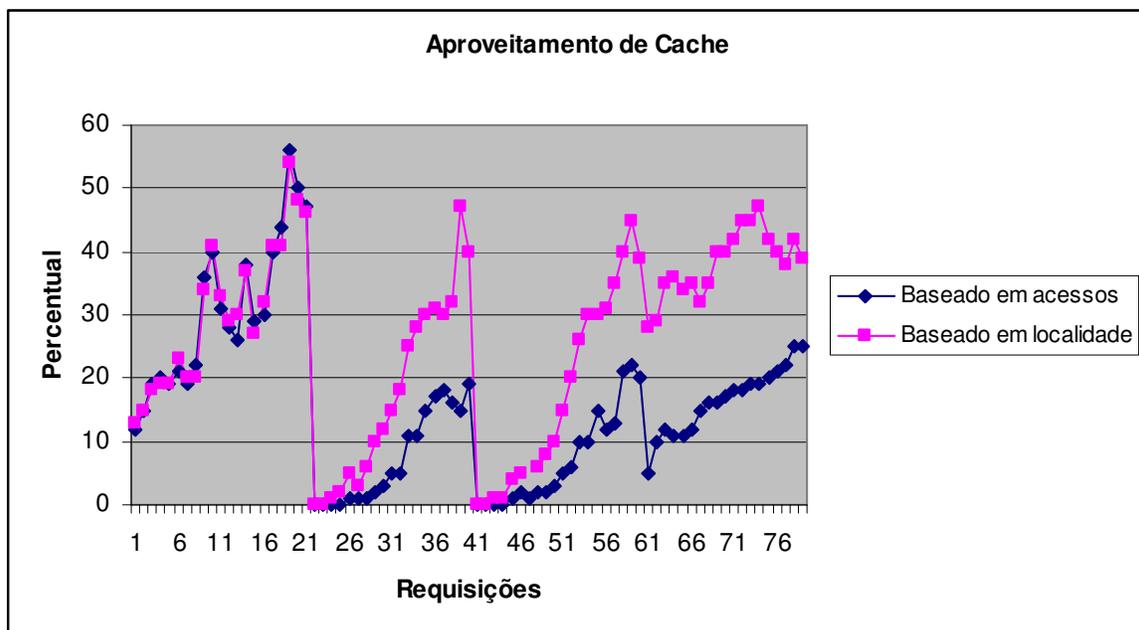


Figura 6.10 - Gráfico comparativo de percentual de aproveitamento dos dados em cache

As primeiras requisições de dados processadas pelos programas não resultam em procedimentos de substituição, visto que 40% do espaço em cache está disponível inicialmente. Após as primeiras

iterações, contudo, o processo de substituição deve ser realizado a cada resposta de uma requisição, pois novas informações devem ser armazenadas no cache.

Verificamos, pela análise do gráfico, que na posição inicial da rede *ad hoc*, os métodos (o baseado em acesso e o baseado em localidade) possuem percentuais de aproveitamento muito semelhantes. Esta semelhança ocorre devido ao fato de que, inicialmente, a escolha baseada na posição não é muito crítica, pois os itens mantidos em cache foram recentemente incluídos e todos possuem, portanto, uma localização próxima da posição corrente da rede. É importante observar que neste ponto é provável que o método baseado somente em acessos seja até mais eficiente.

Entretanto, após a primeira mudança de posição dos dispositivos, pode-se observar uma considerável diferença entre os desempenhos de ambos os métodos, resultando na conclusão de que o método baseado somente em acessos não é adequado quando ocorre mudança de localização, não sendo, portanto, adequado para ambientes móveis sem fio.

Tal constatação é feita tendo como base o fato de que este método mantém em cache os itens com maior quantidade de acessos, o que desfavorece os novos itens, com poucos acessos, porém mais próximos da nova localidade da rede. Dessa forma, ao se efetuar procedimentos de substituição para que sejam armazenadas as novas informações, os itens mais antigos são favorecidos para permanecerem em cache, por possuírem maior quantidade de acessos, o que implica em maior chance de descarte dos itens mais próximos da nova posição da rede (recentemente incluídos) e com maior chance de acesso futuro.

Como citado anteriormente, mantivemos nestes testes um arquivo indicando os elementos recentemente descartados após os procedimentos de substituição, com o intuito de verificar a correteza das escolhas feitas nesses procedimentos realizados por ambos os métodos. A Figura 6.11 abaixo apresenta o gráfico obtido dessa verificação.

De forma semelhante à comparação de percentual de aproveitamento apresentada acima, no momento inicial dos testes (antes de ocorrer qualquer mudança de localização), os comportamentos dos dois métodos são bastante semelhantes. Assim como o primeiro estudo comparativo, a localização dos itens não é um fator crítico neste momento, já que os dados em cache são todos oriundos de respostas de CDLs submetidas nessa posição corrente.

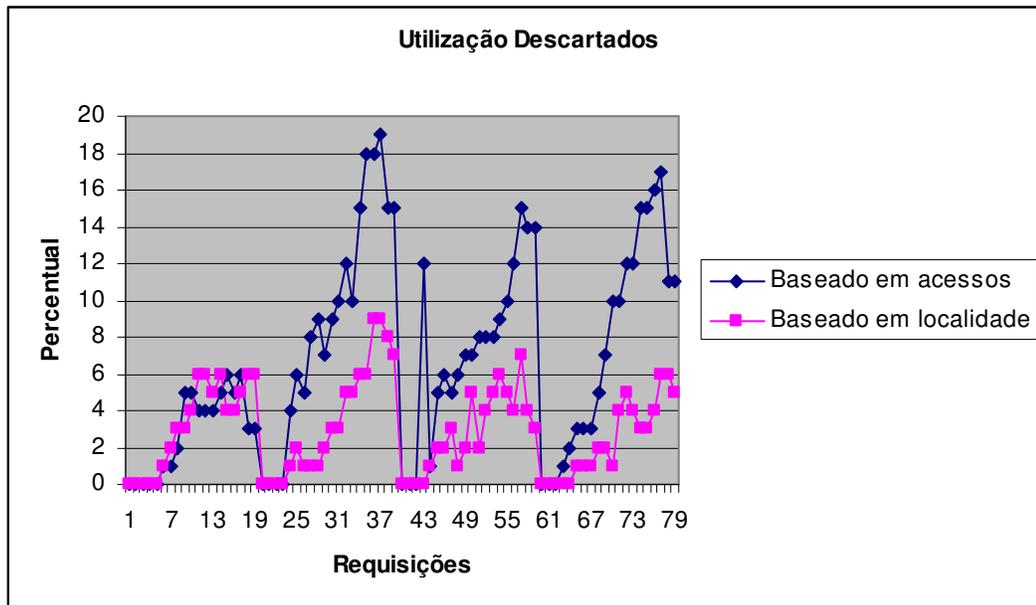


Figura 6.11 - Gráfico comparativo da utilização dos itens recentemente descartados

Entretanto, em um segundo momento, após a locomoção dos dispositivos móveis para uma nova posição, pode-se observar no gráfico da estratégia baseada em acessos, que um maior percentual de itens solicitados em consultas estão sendo encontrados em *clusters* recentemente descartados. Isto significa, portanto, que a escolha dos itens eliminados não está sendo realizada da forma mais adequada.

Essas escolhas equivocadas ocorreram exatamente pelo mesmo motivo exposto na análise do primeiro gráfico, no qual ocorre um favorecimento dos itens que foram muito acessados em uma localização anterior e que possuem, portanto, um maior número de acessos, sendo que tais itens estão agora distantes da nova posição da rede, fato este que os impossibilita de serem utilizados como resposta a uma CDL nesta nova posição. Desse modo, o descarte dos itens, recebidos recentemente do servidor fixo, explica o comportamento do gráfico da Figura 6.11.

6.7. Discussão

Neste capítulo apresentamos o desenvolvimento de um programa computacional utilizado como projeto inicial do mecanismo de gerenciamento de cache proposto e, adicionalmente, realizamos alguns experimentos com o intuito de mostrar a eficiência das estratégias propostas neste trabalho.

Para uma análise geral da validade do nosso método é importante considerar três pontos principais, que caracterizam o mecanismo de gerência de cache apresentado nesta dissertação:

- **A atuação do mecanismo em um servidor móvel**

Conforme discutido na Seção 5.2, a utilização de um servidor móvel, na arquitetura considerada neste trabalho, pode aumentar a taxa de reaproveitamento de dados para os usuários da rede *ad hoc*, pois, com a utilização desse servidor em um equipamento de maior capacidade computacional, é possível aumentar a quantidade de dados disponíveis na rede *ad hoc*, reduzindo o número de requisições ao servidor externo através da rede de longa distância.

- **A manutenção de dados em formato de itens XML ao invés de páginas**

Essa segunda característica foi parcialmente discutida em [56] e [45], onde os autores apresentam vários experimentos com o intuito de avaliar extensivamente o desempenho de gerência de cache por páginas em relação ao gerenciamento de cache semântico, sendo que este último realiza a manutenção e tratamento das informações em cache por unidades semânticas de itens de dados, ao invés de páginas.

Dessa forma, por [56] e [45], fica claro que, para ambientes móveis sem fio, a gerência de cache de baseada em itens de dados é mais eficiente do que a gerência baseada em páginas. Nesse sentido, tomando como base essas constatações, estendemos, neste trabalho, as unidades semânticas em itens XML. Tal extensão torna nosso mecanismo ainda mais eficaz, visto que o cache do servidor móvel possui um formato único para tratamento e transferência de informações.

- **A consideração da localização do dispositivo como critério essencial nas escolhas feitas nas estratégias de substituição e validação**

Finalmente, a consideração da localização do dispositivo como critério essencial nas estratégias de substituição e validação de cache é o fator principal pelo qual nosso mecanismo torna-se mais adequado para ambientes móveis sem fio. Como mostrado na Seção 6.6, a partir de um estudo comparativo entre duas estratégias de substituição, concluímos que, se considerarmos a característica

de localização como critério relevante nessa estratégia, é possível aumentar a taxa de aproveitamento de dados em cache e, conseqüentemente, melhorar a disponibilidade de dados na rede *ad hoc*.

Além disso, nas estratégias de validação implementadas no gerenciador de cache do servidor móvel, a consideração de tal característica de localização também se faz necessária, visto que, nesse trabalho inicial, o tipo de consulta abordada (CDLs) depende do critério de localização especificado.

6.8. Conclusão

Neste capítulo, foi apresentada a implementação do mecanismo de gerenciamento de cache proposto nesta dissertação, bem como foram analisados os resultados obtidos a partir de experimentos realizados com o auxílio do programa computacional implementado.

No capítulo seguinte, apresentamos as conclusões tiradas da realização deste trabalho, suas contribuições e suas perspectivas futuras.

CAPÍTULO 7

CONCLUSÃO

Neste capítulo, apresentamos as conclusões desta dissertação, bem como suas perspectivas futuras.

7.1. Considerações sobre o Mecanismo de Cache Proposto

O alto custo de comunicação de uma rede celular nos motivou a considerar uma arquitetura na qual um mecanismo gerenciador de cache, atuando em um servidor móvel de uma rede *ad hoc*, pode diminuir consideravelmente o fluxo de dados na interface aérea de uma rede celular. Tal diminuição reduz, portanto, o custo geral referente à solicitação de consulta por parte dos usuários móveis da rede *ad hoc*.

A capacidade de mudar constantemente de localização, atribuída aos usuários de sistemas móveis sem fio, adiciona complexidade a esses sistemas, pois surge uma propriedade conhecida por dependência de localização. Verificamos que tal propriedade afeta diretamente as estratégias de substituição e validação de cache e, por esse motivo, consideramos a localização como critério relevante nos processos de gerenciamento de cache propostos nesse trabalho.

Com o intuito de introduzir os problemas de gerência de cache causados pela propriedade de dependência de localização e apresentar soluções para esses problemas, os estudos realizados neste trabalho foram direcionados às consultas dependentes de localização (ou CDLs). Sendo assim, foi possível elaborar estratégias adequadas para atuarem em ambientes móveis sem fio sem, no entanto, invalidar as estratégias já existentes em sistemas de gerenciamento de cache tradicionais.

Desenvolvemos, então, estratégias e algoritmos de validação e substituição de cache que consideram a localização do dispositivo móvel como critério relevante nas escolhas a serem tomadas no mecanismo de gerenciamento de cache.

Através do desenvolvimento de um programa computacional, foi possível realizar comparações entre dois tipos de estratégias de substituição de dados em cache e, após a análise dessas comparações, observamos que os métodos implementados pelo mecanismo de gerência de cache proposto se comportou de forma mais eficiente, ao se efetuar mudança nas posições dos dispositivos.

7.2. Trabalhos Futuros

Como trabalhos futuros, pretendemos adaptar o mecanismo proposto para atuar em qualquer tipo de consulta de leitura, tornando-o mais abrangente, pois, apesar de não invalidar estratégias de gerenciamento de cache tradicionais, o mecanismo proposto fornece métodos que consideram apenas consultas dependentes de localização, abstraindo-se do tratamento de outros tipos de consultas. Consideraremos também consultas de atualização de dados possibilitando, desse modo, fornecer estratégias para garantir a consistência dos dados mantidos em cache.

Além disso, pretendemos integrar o mecanismo de gerência de cache do servidor móvel com o módulo gerenciador de armazenamento FoX deste servidor, de modo que sejam consideradas as restrições de replicação de dados acarretadas a partir dessa funcionalidade.

Finalmente, consideramos como um outro ponto importante a ser realizado numa continuação deste trabalho, a realização de um estudo de caso para ilustrar a utilização das redes *ad hoc* por usuários com interesse comum e, desse modo, validar de maneira mais precisa a utilização do gerenciador de cache em um servidor móvel.

7.3. Considerações Finais

Neste trabalho apresentamos, como principal contribuição, um mecanismo de gerência de cache que atua em um servidor móvel de uma rede *ad hoc*, desenvolvido para funcionar em um ambiente móvel sem fio, no qual usuários móveis possuem capacidade de mudar de localização frequentemente. Esse mecanismo mostrou-se capaz de melhorar a disponibilidade de dados na rede *ad hoc* e diminuir, conseqüentemente, o fluxo de dados na interface aérea de uma rede celular, reduzindo, portanto, o custo geral do processamento de CDLs.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1]. APACHE. **Xerces**. Disponível em: <<http://xml.apache.org/xerces-c/index.htm>>.
- [2]. BADRINATH, B. R.; IMIELINKSI, T. Replication and Mobility. In *Proceedings of 2nd WORKSHOP ON THE MANAGEMENT OF REPLICATED DATA*, pp. 9-12, 1992.
- [3]. BADRINATH, B.R.; PHATAK, S.H. An Architecture for Mobile Databases. **Technical Report** DCS-TR-351, Department of Computer Science, Rutgers University. 1999.
- [4]. BARBARA, D. Mobile Computing and Databases: A Survey. **IEEE Transactions Knowledge and Data Engineering**, vol. 11, no. 1, pp. 108-117, Jan./Feb. 1999.
- [5]. BARBARA, D., IMIELINKI, T. Sleepers and Workaholics: Caching Strategies for Mobile Environments. In: *proceedings of ACM SIGMOD*, pp. 1-12, 1994.
- [6]. BLACK, U. **Second Generation Mobile & Wireless Networks**. Prentice Hall Series in Advanced Communications Technologies – Prentice Hall, 1998.
- [7]. BUKHRES, O.; MORTON, S.; MOSMAN, M. Mobile Computing Architecture for a Battlefield Environment. In: *Proceedings of the INTERNATIONAL SYMPOSIUM ON COOPERATIVE DATABASE SYSTEMS FOR ADVANCED APPLICATIONS*, 1996.
- [8]. CAO, G. Proactive Power-Aware Cache Management for Mobile Computing Systems, **IEEE Transactions on Computers**, v. 51, no. 6, pp. 608-621, JUN 2002.
- [9]. CHEN, L. AND RUNDENSTEINER, E. A. ACE-XQ: A CachE-aware XQuery Answering System, ACM SIGMOD ASSOCIATED WORKSHOP ON THE WEB AND DATABASES (WebDB'02), pp 31-36, Madison, Wisconsin, June 2002.

- [10]. CHEN, L. AND RUNDENSTEINER, E. A., WANG, S. Xcache – A Semantic Caching System for XML Queries. ACM SIGMOD'02, Software System Demonstration Paper, Madison, Wisconsin, pp 618, June 2002.
- [11]. CHEN, L., WANG, S., CASH E., RYDER B., HOBBS I., RUNDENSTEINER, E.A. A Fine-Grained Replacement Strategy for XML Query Cache, WORKSHOP ON WEB INFORMATION AND DATA MANAGEMENT (WIDM'02), Nov. 2002.
- [12]. DASBIT, S.; MITRA, S. Challenges of computing in mobile cellular environment – a survey. **Computer Communications**, vol. 26, issue 18, pp. 2090-2105, Dec. 2003.
- [13]. DATTA, A., DUTT, K., THOMAS, H., VANDERMEER, D. A Comparative Study of Alternative Middle Tier Caching Solutions to Support Dynamic Web Content Acceleration In: *Proceedings of VLDB CONFERENCE*, Rome, Italy, Sep. 2001.
- [14]. DORNAN, A. **Wireless Communication – O Guia Essencial da Comunicação sem Fio**. Tradução: Fábio Freitas – Rio de Janeiro, Ed. Campus, 2001.
- [15]. DUNHAM, M. H.; KUMAR, V. Location Dependent Data and its Management in Mobile Databases. In: *Proceedings of DEXA WORKSHOP*, pp. 414-419, Vienna, Austria, Aug. 1998.
- [16]. FEGARAS, L.; LEVINE, D.; BOSE, S.; CHALUVADI, V. Query Processing of Streamed XML Data. In: *Proceedings of 11th INTERNATIONAL CONFERENCE ON INFORMATION AND KNOWLEDGE MANAGEMENT*, pp. 126-133, USA, 2002.
- [17]. FLANKLIN, M. et al. Transactional Client-Server Cache consistency: Alternatives and Performance, **TTODS**, vol. 22, no. 3, 1997.
- [18]. FORMAN, G.H.; ZAHORDAN, J. The challenges of mobile computing. **IEEE Computer**, v. 27, issue:4, pp 38 –47, Apr 1994.

- [19]. FRANKLIN, M. J., CAREY, M.J. Client-Server Caching Revisited. In: *Proceedings of INT'L WORKSHOP ON DISTRIBUTED OBJECT MANAGEMENT (IWDOM)*, pp. 57-78, Edmonton, Canada, Aug. 1992.
- [20]. FRANKLIN, M. J., CAREY, M.J., LIVNY, M. Local Disk Caching for Client-Server Database Systems. In: *Proceedings of the 19th VLDB (International Conference on Very Large Data Bases)*, pp. 641-655, Dublin, Ireland, Aug. 1993.
- [21]. FRIEDMAN, R. Caching Web Services in Mobile Ad-hoc Networks: Opportunities and Challenges. In: *Proceedings of 2nd ACM INTERNATIONAL WORKSHOP ON PRINCIPLES OF MOBILE COMPUTING*, pp 90-96, França, 2002
- [22]. GÖK, H.G. **Processing of Continuous Queries From Moving Objects in Mobile Computing Systems**. Thesis (degree of Master) – Bilkent University, Jan. 1999.
- [23]. HAAS, Z. J.; Liang, B. Ad Hoc Mobility Management with Randomized Database Groups. **IEEE International Conference on Communication**, vol. 3, pp. 1756-1762, Jun 1999.
- [24]. HELAL, A.; HAMMER, A.; ZHANG, J. A Three-tier Architecture for Ubiquitous Data Access. In: *Proceedings of 1st ACS/IEEE INTERNATIONAL CONFERENCE ON COMPUTER SYSTEMS AND APPLICATIONS*, Beirut, Lebanon, pp. 177-180, 2001.
- [25]. HU, Q. L.; LEE, D. L; LEE, W. C. Data Delivery Techniques in Asymmetric Communication Environments, Aug 1999.
- [26]. HUANG, Y.; SISTLA, P.; WOLFSON, O. Data Replication for Mobile Computers. In: *Proceedings of ACM SIGMOD INTERNATIONAL CONFERENCE ON MANAGEMENT DATA*, pp. 13-24, 1994.
- [27]. IMIELINSKI, T.; BADRINATH, B. R. Querying in highly mobile distributed environments. In: *Proceedings of 18th INT'L CONF VERY LARGE DATA BASES*, Vancouver, B. C., Canada, Aug. 1992.

- [28]. IYENGAR, A.; CHALLENGER, J. Improving Web Server Performance by Caching Dynamic Data. In: *Proceedings of the USENIX SYMPOSIUM ON INTERNET TECHNOLOGIES AND SYSTEMS*, Dec. 1997.
- [29]. JAGADISH, H.V.; AL-KHALIFA, S., ET AL. TIMBER: A native XML database. **VLDB Journal**, 11(4), 2002.
- [30]. JAGADISH, H.V.; e al. Timber: A native XML Database. **VLDB Journal**, 11(4), Jun. 2002.
- [31]. JENSEN, C.S. E AL. Location-based services: a database perspective. In: *Proceedings of the 8th SCANDINAVIAN RESEARCH CONFERENCE ON GEOGRAPHICAL INFORMATION SCIENCE*, pp. 59-68, Jun 2001.
- [32]. JING, J.; BUKHRES, O.; ELMARGARMID, A. K.; ALONSO, E. Bit-Sequences: A new cache invalidations method in mobile environments. **MONET**, 2(II), 1997.
- [33]. JING, J.; HELAL, A.; ELMAGARMID, A. Client-Server Computing in Mobile Environments. **ACM Computing Surveys**. Vol. 31, no. 2, 1999
- [34]. JUNG, I.; YOU, Y.; LEE, J.; KIM, K. Broadcasting and Caching Policies for location-dependent queries in urban areas, In: *Proceedings of the 2nd INTERNATIONAL WORKSHOP ON MOBILE COMMERCE*, Atlanta, Georgia, USA, Sep 2002.
- [35]. KANNE, C; MOERKOTTE, G. Natix – Efficient Storage of XML Data. In: *Proceedings of ICDE CONF.*, poster abstract, p 198, Calif., USA, Mar 2000.
- [36]. KAYAN, E.; ULUSOY, O. “Real-time transaction management in mobile computing systems”. In: *Proceedings of 6th INTERNATIONAL CONFERENCE ON DATABASE SYSTEMS FOR ADVANCED APPLICATIONS*, pp. 127-134, 1999.
- [37]. KHOSHAFIAN, S.; COPELAND, G. Object Identity In: *Proceedings of INT. CONF. ON OOPSLA*, pp.406-416, Sep. 1986.

- [38]. KHOSHAFIAN, S.;VANDURIEZ, P. Sharing Persistence and Object-Orientation: A Database Perspective. In: *Proceedings of* INT. WORKSHOP ON DATABASE PROGRAMMING LANGUAGES, pp. 181-205, Sep. 1987.
- [39]. KIM, S. W.; KIM, Y. H.; LEE, J.; LIM, HAECHELL. Developing a Native Storage for XML Repository System in Main Memory. Korea, 2002.
- [40]. LAM, FRANKLY; LAM, NICOLE; WONG, R. Efficient Synchronization for Mobile XML Data. ACM 2002. In: *Proceedings of* the 11th INT. CONFERENCE ON INFORMATION AND KNOWLEDGE MANAGEMENT, pp 153-160, USA 2002
- [41]. LAM, K.; ULUSOY, O.; LEE, T.S.H.; CHAN, E.; LI, G. An efficient method for generating location update for processing of location-dependent continuous queries. In: *Proceedings of* 17th. INTERNATIONAL CONFERENCE ON DATABASE SYSTEMS FOR ADVANCED APPLICATIONS, pp. 218-225, 2001.
- [42]. Leal, D; Machado, J.; Andrade, R. Um Mecanismo de Gerência de Cache para um Servidor de Dados Móvel. II WORKSHOP DE TESES E DISSERTAÇÕES EM BANCOS DE DADOS. **Anais...** pp. 131-136, Manaus, Outubro,2003.
- [43]. LEE, D.L.; XU, JIANLIANG; ZHENG, B.; LEE, W. Data Management in location-dependent information services. **IEEE Pervasive Computing**, vol. 1, issue 3, pp. 65-72, 2002.
- [44]. LEE, K.C.K.; LEONG, H.V.; SI,A. Semantic Query Caching in a mobile environment, **ACM SIGMOBILE Mobile Computing and Communications Review**, v. 3, issue 2, Apr 1999.
- [45]. LEONG, H.V.; SI, A. On Adaptive Caching in Mobile Databases. In: *Proceedings of* ACM SYMPOSIUM ON APPLIED COMPUTING. April, 1997.
- [46]. LUBINSKY, A. Database Security meets Mobile Requirements, In: *Proceedings of* INTERNATIONAL SYMPOSIUM ON DATABASE TECHNOLOGY SOFTWARE ENGINEERING, WEB and Cooperative Systems, Baden., 2000

- [47]. MARTIN, C.P.; RAMAMRITHAM, K. Recovery guarantees in mobile systems. In: *Proceedings of the ACM INTERNATIONAL WORKSHOP ON DATA ENGINEERING FOR WIRELESS AND MOBILE ACCESS*, Aug, 1999.
- [48]. MAURO, R.C. **Aspectos de Gerência de Objetos Persistentes: A implementação do GOA++**. 1998. 116f. Dissertação (Mestrado em Engenharia de Sistemas e Computação) – COPPE, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 1998.
- [49]. MAURO, R.C.; MATOSO, M.L.Q. ET AL. GOA++: Tecnologia, implementação e extensões aos serviços de gerência de objetos. XII SIMPÓSIO BRASILEIRO DE BANCOS DE DADOS. **Anais...** pp. 272-286, Fortaleza, Outubro, 1997.
- [50]. MCDONALD, A. B.; ZNATI, T. A mobility based framework for Adaptive Clustering in Wireless Ad Hoc Networks. **IEEE Journal on Selected Areas in Communication**, Vol. 17, No. 8, Aug 1999.
- [51]. PINHEIRO, C. **Uma Estratégia de Indexação para Dados XML**. Dissertação (Mestrado em Ciência da Computação), Universidade Federal do Ceará, Fortaleza, 2004.
- [52]. PITOURA, E.; BHARGAVA, B. Building Information System for Mobile Environments. In: *Proceedings of the 3rd INTERNATIONAL CONFERENCE ON INFORMATION AND KNOWLEDGE MANAGEMENT*, pp. 371-378, Guithesburg, MD, November 1994.
- [53]. PITOURA, E.; BHARGAVA, B. Dealing with Mobility: Issues and Research Challenges. **Technical Report TR-97-070**, Department of Computer Sciences, Pardue University, 1993.
- [54]. QUAN, L. P., CHEN, L. AND RUNDENSTEINER, E. A. Argos: Efficient Refresh in an XQL-Based Web Caching System. **ACM SIGMOD ASSOCIATED WORKSHOP ON THE WEB AND DATABASES (WEBDB 2000)**, pp 23-28, Dallas, Texas, May 2000.

- [55]. REN, Q.; DUNHAM, M.H. Using Semantic Caching to Manage Location Dependent Data in Mobile Computing. In: *Proceedings of 6th ANN. ACM/IEEE CONF. MOBILE COMPUTING AND NETWORKING (MobiCom 2000)*, pp. 210-221, Aug. 2000.
- [56]. REN, Q.; DUNHAM, M.H.; KUMAR, V. Semantic Caching and Query Processing, **IEEE Transactions on Knowledge and Data Engineering**, v.15, n. 1, pp.192-210, Jan/Feb 2003.
- [57]. REN,Q.; DUNHAM, M.H. Using clustering for effective management of a semantic cache in mobile computing. In: *Proceedings of the ACM INTERNATIONAL WORKSHOP ON DATA ENGINEERING FOR WIRELESS AND MOBILE ACCESS*, Aug. 1999.
- [58]. SEYDIM, A.Y.; DUNHAM, M.; KUMAR, V. An Architecture for Location Dependent Query Processing. In: *Proceedings of 12th International Workshop on Database and Expert System Applications*, pp. 549-555, 2001.
- [59]. SISTLA, A.P.; WOLFSON, O.; HUANG, Y. Minimization of Communication Cost Throught Caching in Mobile Environments, **IEEE Transactions on Parallel and Distributed Systems**, v. 9, no. 4, Apr 1998.
- [60]. TANENBAUM, A. S. **Sistemas Operacionais Modernos**. Tradução: Nery Machado Filho – Rio de Janeiro, Ed. Prentice-Hall do Brasil, 1992.
- [61]. TASMAN, M.P. **Protocols and Caching Strategies in Support of Internetwork Mobility**. Thesis (Degree of Doctor), University of Wisconsin, Madison, 1994.
- [62]. TERRY, D. B.; RAMASUBRAMANIAN, V. Caching XML Web Services. Microsoft Research, May 2003.
- [63]. TSUKAMOTO, M.; KADOBAYASHI, R.; NISHIO, S. Strategies for Query Processing in Mobile Computing. In **Mobile Computing**, pp. 595-620. Kluwer Academic Publishers, 1996.

- [64]. VALDURIEZ, O.; ÖZSU, M., T. **Princípios de Sistemas de Banco de Dados Distribuídos**. Ed. Campus. Tradução: Vandenberg D. de Souza – Rio de Janeiro, 2001.
- [65]. WOLFSON, O.; XU, B.; CHAMBERLAIN, S.; JIANG, L. Moving Objects Databases: Issues and Solutions. In: *Proceedings of the 10th INTERNATIONAL CONFERENCE ON SCIENTIFIC AND STATISTICAL DATABASE MANAGEMENT*, pp. 112-122, 1998.
- [66]. World Wide Web Consortium (W3C). **Document Object Model (DOM)**. Disponível em: <http://www.w3.org/TR/REC-DOM-Level-1/>. Acesso em: Janeiro 2004.
- [67]. World Wide Web Consortium (W3C). **Extensible Markup Language (XML)**. Disponível em: <http://www.w3.org/XML/>. Acesso em: Janeiro 2004.
- [68]. World Wide Web Consortium (W3C). **XML Path Language (XPath)**, 2000. Disponível em: <http://www.w3.org/TR/XPath/>. Acesso em: Fevereiro 2004.
- [69]. WU, K.; YU, P.S.; CHEN, M. Energy-Efficient Mobile Cache Invalidation. **Distributed and Parallel Databases**, v. 6, pp 351-372, 1998.
- [70]. XU, J.; TANG, X.; LU, D.L.; HU, Q.L. Cache Coherence in Location-dependent Information Services for mobile environment. In: *Proceedings of FIRST INT'L CONF. MOBILE DATA ACCESS (MDA'99)*, pp. 182-193, Dec. 1999.
- [71]. YUEN, J., CHAN, E., LAM, K., LEUNG, H. Cache Invalidation Scheme for Mobile Computing Systems with real-time Data. **ACM SIGMOD Record**, Dec. 2000.
- [72]. ZHENG, B.; XU, J.; LEE, D.L. Cache Invalidation and replacement strategies for location-dependent data in mobile environments, **IEEE Transactions on Computers**, v. 1, issue 10, pp. 1141-1153, Oct 2002.
- [73]. ZHOU, L.; HAAS, Z., J. Securing Ad hoc Networks, **IEEE Network**, pp. 24-30, Dec. 1999.

- [74]. ZHU, H. YANG, T. Cachuma: Class-based Cache Management for Dynamic Web Content. Technical Report TRCS00-13, Dept. of Computer Science, The University of California at Santa Barbara, Jun., 2000.