

PROCESSAMENTO DE CONSULTAS ESPACIAIS BASEADO EM CACHE SEMÂNTICO DEPENDENTE DE LOCALIZAÇÃO

Heloise Manica^{1,2}, Murilo S. de Camargo³, Cristina Dutra de Aguiar Ciferri⁴
and Ricardo Rodrigues Ciferri⁴

¹*Departamento de Informática e Estatística, Universidade Federal de Santa Catarina, Florianópolis, SC, Brasil - heloise@inf.ufsc.br*; ²*Universidade Paranaense, Paranavaí, PR, Brasil*; ³*Departamento de Ciência da Computação, Universidade de Brasília, Brasília, DF, Brasil - murilo@cic.umb.br*; ⁴*Departamento de Informática, Universidade Estadual de Maringá, Maringá, PR, Brasil - cdac@uol.com.br, rrciferri@uol.com.br*

Abstract: A special class of applications, which connects spatial database and mobile computing, requires query processing whose results depend on the geographic location of the mobile unit. Dealing with these location-dependent queries is the focus of this work. This paper presents a spatial query processing model based on location-dependent semantic cache. In addition, this paper proposes a solution for the reorganization of the cached semantic segments.

Key words: spatial databases; mobile computing; spatial query processing; cache management; location-dependent semantic cache.

1. INTRODUÇÃO

Serviços dependentes de localização estão presentes em uma classe especial de aplicações que integra banco de dados espaciais e computação móvel. Nesses serviços, a resposta de uma consulta varia conforme a posição geográfica de uma unidade móvel. Tal consulta é denominada de consulta dependente de localização (CDL), a qual é frequentemente realizada a partir de equipamentos móveis, tais como *notebook* e *palmtop* (Dunham and Kumar, 1998; Xu et al., 1999; Manica and Camargo, 2004).

Um exemplo típico de CDL é encontrado em aplicações de páginas amarelas (Ren and Dunham, 2000). Nessas aplicações, uma unidade móvel deseja recuperar serviços próximos à sua posição corrente, tais como informações sobre hotéis, restaurantes e cinemas. À medida que a unidade móvel se desloca, o conjunto resposta da consulta é alterado.

Para responder uma CDL, primeiramente é feito um mapeamento da CDL para uma consulta espacial tradicional. Este mapeamento é realizado com base na posição corrente da unidade móvel, a qual pode ser fornecida por um sistema de posicionamento por satélite.

Devido às restrições de capacidade de armazenamento e processamento comumente presentes nas unidades móveis, a consulta mapeada é processada por um sistema gerenciador de banco de dados espacial localizado em um servidor remoto. Esta abordagem segue o modelo cliente-servidor em uma rede sem fio, no qual a unidade móvel envia a consulta espacial para o servidor remoto, o servidor processa a consulta espacial e depois devolve a resposta da consulta para a unidade móvel.

Apesar da corretude deste modelo, a transmissão da consulta espacial e da sua resposta pela rede sem fio introduz alguns fatores limitantes. A rede sem fio possui largura de banda reduzida e também é vulnerável a desconexões. Outro fator limitante refere-se ao consumo de bateria dos equipamentos móveis. Quanto maior a comunicação com o servidor, maior o consumo de bateria.

Uma solução para estas limitações consiste na utilização de um cache semântico na unidade móvel (Lee et al., 1999; Ren and Dunham, 1999; Ren, 2000; Ren et al., 2003). Nesta solução, as respostas das consultas obtidas do servidor são armazenadas em segmentos. Cada segmento possui um conjunto de tuplas, uma condição de seleção que descreve as tuplas agrupadas dentro do segmento e um valor de substituição. Cada tupla no cache semântico é associada a um único segmento.

Este artigo apresenta uma proposta de processamento de consultas espaciais em um cache semântico dependente de localização (CSDL) e também discute sobre a reorganização de segmentos semânticos devido à execução destas consultas espaciais. O restante deste artigo está estruturado da seguinte forma. A seção 2 sumariza trabalhos correlatos. Conceitos relacionados com CSDL são descritos na seção 3. A seção 4 apresenta a proposta de processamento de consultas espaciais em um CSDL, enquanto a seção 5 apresenta a proposta para a reorganização dos segmentos semânticos. O artigo é concluído na seção 6.

2. TRABALHOS CORRELATOS

Já existem trabalhos relacionados ao gerenciamento de cache para sistemas dependentes de localização (Dunham and Kumar, 1998; Lee et al., 2002; Zheng et al., 2002; Xu et al., 2003).

Em Dunham and Kumar (1998) e Lee et al. (2002), os autores introduzem conceitos, discutem o gerenciamento de dados em sistemas dependentes de localização e destacam novas áreas de pesquisa que surgiram com a necessidade de acesso a dados sob-demanda, de radiodifusão e de caching de dados.

Três novas estratégias para a invalidação de cache para sistemas dependentes de localização são apresentadas em Zheng et al. (2002). A primeira delas, denominada *poligonal endpoints*, prioriza a precisão das respostas das consultas a um custo de espaço elevado. A segunda proposta, denominada *approximate circle*, diminui o requisito de espaço, porém propicia uma menor precisão das respostas. Finalmente, a estratégia *caching-efficiency-based* visa balancear os requisitos apresentados nas estratégias anteriores.

Xu et al. (2003) propõem três modelos para a invalidação de cache dependente de localização: *bit vector with compression*, *grouped bit vector with compression* e *implicit scope information*. O primeiro modelo é adequado para sistemas pequenos, sendo necessário o uso de métodos de compressão para ser aplicado a sistemas maiores. Já o segundo método é voltado para grandes sistemas. O terceiro método, por sua vez, armazena os dados conforme seções lógicas para diferentes áreas válidas. Os autores investigam e validam os modelos propostos por meio de um estudo analítico.

Outra área relacionada a este artigo consiste na proposta de modelos de cache semântico. Estão presentes na literatura trabalhos relacionados ao tratamento de cache semântico para computação móvel (Dar et al., 1996; Ren and Dunham, 2000). Entretanto, o relacionamento espacial no gerenciamento dos segmentos semânticos não é completamente explorado nestes trabalhos.

O primeiro trabalho a considerar o fator distância semântica em sua política de substituição de cache é o de Dar et al. (1996). Neste trabalho, os autores utilizam o cálculo *Manhattan distance* para calcular a distância de cada segmento semântico da posição atual do usuário. Diferentemente de Dar et al., 1996, neste artigo nós utilizamos a política ASCR (*Adaptive Semantic Cache Replacement approach*) proposta em Manica et al. (2004).

Ren and Dunham (2000) propõem um modelo de cache semântico para aplicações dependentes de localização. No entanto, o relacionamento espacial é utilizado somente na política de substituição de segmentos. Para agilizar a localização de segmentos candidatos para responder uma consulta,

Ren and Dunham (2000) adotam heurísticas. Na nossa proposta, o relacionamento espacial é utilizado no processamento da consulta espacial, apresentado neste artigo, e na política de substituição, descrita em Manica et al. (2004). A estratégia proposta neste artigo baseia-se no uso de estruturas de indexação para agilizar a localização dos segmentos semânticos.

3. CACHE SEMÂNTICO DEPENDENTE DE LOCALIZAÇÃO

Um CSDL armazena no disco do cliente (i.e., na unidade móvel) as respostas das últimas CDL que foram executadas, bem como as suas áreas geográficas relacionadas. Desta forma, novas CDL podem reaproveitar as respostas no CSDL, visando evitar ou diminuir a transmissão de dados entre a unidade móvel e o servidor remoto. As respostas das CDL são organizadas em segmentos semânticos, ao invés da organização tradicional dos dados em páginas de disco.

Este artigo estende Manica et al. (2004), no qual é proposto um novo modelo para CSDL. Esse modelo possui um esquema de organização baseado em regiões pré-definidas. Ademais, uma CDL é definida da seguinte forma nesse modelo. Para um banco de dados $D = \{R_i\}$, onde R_i consiste de um conjunto de relações, uma CDL consiste de uma tupla (Q_R, Q_P, Q_J, Q_C) , sendo que $Q_R \in D$ refere-se à relação pesquisada no banco de dados, Q_P representa o predicado da CDL, Q_J refere-se ao MBR (*Minimum Bounding Rectangle*) da área geográfica da consulta e Q_C representa a resposta da consulta.

O modelo para CSDL proposto em Manica et al. (2004) é composto de três partes. A primeira parte é relativa ao conteúdo do CSDL e possui as respostas das CDL, as quais são agrupadas em segmentos semânticos de tamanho variável. A segunda parte é composta por um índice de segmentos semânticos, o qual mantém informações descritivas e também de armazenamento físico para cada segmento armazenado no CSDL. O índice é representado pela tupla $(S, S_R, S_P, S_A, S_C, S_{ts}, S_G)$, onde S é o identificador do segmento, S_R refere-se à relação envolvida na consulta, S_P representa a condição de seleção no predicado da consulta, S_A é um par de intervalos $([x_1, x_2], [y_1, y_2])$ que representa a área geográfica do segmento (i.e., o seu MBR), S_C indica o endereço do segmento no CSDL, S_{ts} é uma marca de tempo e S_G identifica a região (i.e., grupo) que o segmento pertence. Por fim, a terceira parte consiste de uma estrutura de grupo que identifica os segmentos semânticos contidos em cada uma das regiões pré-definidas.

4. PROCESSAMENTO DE CONSULTAS ESPACIAIS

Esta seção apresenta a nossa proposta para o processamento de consultas espaciais em caches semânticos dependentes de localização. O processamento de consultas proposto é composto por duas etapas. A etapa de seleção de segmentos candidatos é descrita na seção 4.1, ao passo que a etapa de divisão e execução da consulta é detalhada na seção 4.2.

4.1 Seleção de Segmentos Candidatos

Esta etapa visa a obtenção do conjunto de segmentos semânticos candidatos a responder à consulta (i.e., CjSC). A vantagem desta estratégia é evitar a varredura completa de toda a estrutura de segmentos para selecionar as respostas que satisfazem à consulta.

Para uma consulta Q , inicialmente são analisados dois tipos de afinidade para verificar se Q pode ser respondida ou não pelo conteúdo em cache: a relação consultada e a posição geográfica. Primeiramente, são selecionados os segmentos que possuem a mesma relação da consulta, ou seja, os segmentos nos quais $S_R = Q_R$. Em seguida, são verificados quais destes segmentos intersectam a janela de consulta, isto é, para quais segmentos $S_A \cap Q_J \neq \emptyset$.

Após a verificação da relação consultada e da posição geográfica, deve-se analisar um terceiro tipo de afinidade: a relação entre os predicados. Para cada um dos segmentos candidatos, deve-se investigar a relação existente entre o predicado do segmento (i.e., S_p) e o predicado da consulta (i.e., Q_p). Caso os predicados estejam relacionados, ou seja, caso $S_p \rightarrow Q_p$, o segmento é selecionado, desde que ele pode contribuir com a resposta da consulta. Em contrapartida, se $Q_p \cap S_p = \emptyset$, o segmento é descartado.

Como resultado desta etapa, CjSC contém o conjunto de segmentos semânticos candidatos a responder à consulta. Apenas os segmentos que pertencerem a CjSC serão analisados na etapa de divisão e execução da consulta (seção 4.2).

Como exemplo, considere a consulta “Selecione os hotéis cuja distância da unidade móvel é menor do que 5 km e que possuam preços de diária menores do que R\$ 100,00” executada por uma unidade móvel na posição $UM_p = (-15, 45)$. A relação Hotel é definida como Hotel (id, nome, diária, geometria). Essa consulta é representada por:

```
Q:  SELECT *
    FROM Hotel
    WHERE  Hotel.diária < 100 AND
          Distance (UMp, Hotel.geometria) < 5
```

Com base na localização da unidade móvel e na distância explícita no predicado, determina-se a janela de consulta Q_J . Analisando-se a Figura 1, dentre todos os segmentos em cache, verifica-se que os segmentos S_1 , S_3 , S_6 e S_7 estão relacionados com a relação Hotel, mas somente S_1 , S_3 e S_7 intersectam Q_J . Assim, inicialmente $C_{jSC} = \{S_1, S_3, S_7\}$. Apesar do segmento S_4 intersectar Q_J , ele não é selecionado como candidato, pois não se refere à relação Hotel. Por outro lado, o segmento S_6 se refere à relação Hotel, porém não intersecta Q_J . Para verificar a interseção dos segmentos com a janela de consulta, pode-se aplicar um algoritmo de *Intersection Range Query* em um índice espacial (Gaede and Günther, 1998; Ciferri and Salgado, 2000; Ciferri, 2002; Ciferri et al., 2003).

Dando continuidade ao exemplo, suponha que os segmentos semânticos S_1 e S_3 armazenem os hotéis cujos valores de diária são menores do que R\$ 50,00 e R\$ 150,00, respectivamente, e que o segmento semântico S_7 armazene os hotéis cujos valores de diária são maiores do que R\$ 200,00. Neste exemplo, apenas os segmentos S_1 e S_3 são selecionados como candidatos, desde que o predicado de S_7 não contribui com a resposta (i.e., $Q_P \cap S_{7P} = \emptyset$). Assim, o conjunto $C_{jSC} = \{S_1, S_3\}$ é o conjunto a ser analisado na próxima etapa de divisão e execução da consulta.

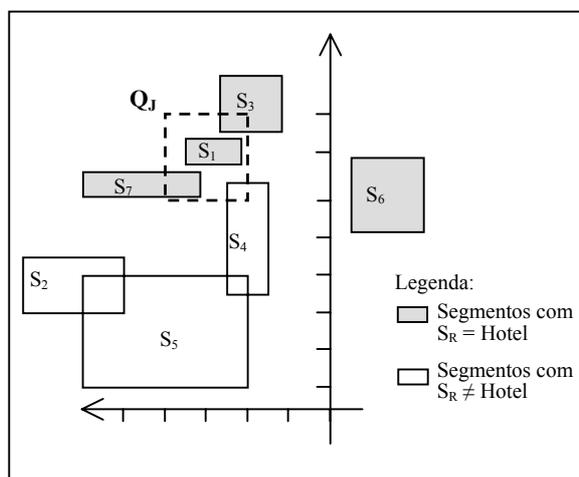


Figura 1. Segmentos no cache semântico.

4.2 Divisão e Execução da Consulta

Nos modelos de cache semântico, uma consulta Q é decomposta em duas partes disjuntas: uma parte que é executada no cache do cliente (i.e., *probe query*), e outra parte que é solicitada ao servidor (i.e., *reminder query*). Esse

processo é conhecido como *query trimming* (Keller and Basu, 1996; Ren et al., 2003). Assim como nos modelos de cache semântico, no processamento de consultas proposto neste artigo, a consulta é dividida em duas partes. Entretanto, diferentemente de outros modelos, nesta proposta a divisão da consulta também é baseada no relacionamento geográfico existente entre a consulta e os segmentos.

O resultado de uma consulta pode ser obtido através de vários segmentos candidatos, isto é, vários segmentos podem contribuir para se obter uma resposta parcial ou até mesmo total para a consulta. A etapa de seleção de segmentos candidatos descrita na seção 4.1 obtém o conjunto C_jSC . Nesta etapa de divisão e execução da consulta, os segmentos do conjunto C_jSC são analisados para definir a parte da resposta da consulta que está em cache e a parte da resposta da consulta que deve ser obtida no servidor. Assim, para cada segmento $S_i \in C_jSC$, são aplicadas as seguintes etapas:

- a) Definir a área de interseção I_i entre S_{iA} e Q_j , isto é, $I_i = S_{iA} \cap Q_j$;
- b) Verificar a relação existente entre os predicados S_{iP} e Q_p . Se $Q_p \supset S_{iP}$, ir para a etapa *c*. Se $Q_p \subseteq S_{iP}$, ir para a etapa *e*;
- c) Enviar uma consulta *amending query* para o servidor com o predicado $Q_p - S_{iP}$ na área I_i e armazenar o resultado em AQS_i ;
- d) Atualizar APQ de forma que $APQ \leftarrow APQ + AQS_i$;
- e) Adicionar o valor de I_i ao vetor X ;
- f) Executar a consulta Q na área I_i e armazenar o resultado em AS_i ; e
- g) Adicionar o resultado parcial obtido em AS_i a APQ, isto é, $APQ \leftarrow APQ + AS_i$.

Primeiramente, na etapa *a* define-se a área de interseção entre a janela de consulta (Q_j) e o segmento (S_{iA}). Em seguida, na etapa *b* verifica-se o relacionamento existente entre o predicado do segmento (S_{iP}) e o predicado da consulta (Q_p), que pode resultar em duas possibilidades. A primeira possibilidade considera o fato do segmento semântico responder parcialmente à consulta, isto é, $Q_p \supset S_{iP}$. Neste caso, uma consulta denominada *amending query* com a parte do predicado que não está em S_{iP} é enviada ao servidor na etapa *c*. Após retornar do servidor, o resultado de AQS_i é adicionado a APQ na etapa *d*. Já a segunda possibilidade considera o fato de que o predicado do segmento satisfaz totalmente o predicado da consulta, isto é, $Q_p \subseteq S_{iP}$. Neste caso, a execução é realizada ignorando-se as etapas *c* e *d*. Na seqüência, a etapa *e* adiciona a área de interseção I_i , que já foi consultada em cache, ao vetor X . As etapas *f* e *g* finalizam o processo, executando Q na área I_i e incrementando o resultado de APQ.

Após todos os segmentos de C_jSC terem sido pesquisados, APQ contém a parte da resposta da consulta Q em cache e X armazena as áreas geográficas que já foram pesquisadas em cache. Caso $X \neq Q_j$, a *reminder query*, executada no servidor, irá retornar toda tupla que estiver dentro da

janela de consulta menos a área que já foi determinada em cache, isto é, $Q_J \wedge \neg X$.

O vetor X deve ser transmitido para o servidor junto com a *reminder query*. Com base neste vetor, o servidor tem conhecimento das áreas já pesquisadas em cache. Adicionalmente, o vetor X também é utilizado pelo servidor para verificar a validade dos dados selecionados em cache. Segundo o nosso conhecimento, este aspecto não é tratado em nenhum trabalho proposto na literatura. A transmissão do vetor X não sobrecarrega o tráfego na rede, já que o vetor X armazena a geometria aproximada de retângulos.

Para melhor explicar a etapa de divisão e execução da consulta, considere o exemplo discutido na seção 4.1. Desde que $C_jSC = \{S_1, S_3\}$, estes dois segmentos são analisados para que os resultados parciais obtidos em cada segmento formem a *probe query*.

Na análise de S_1 , nota-se que o seu predicado (S_{1P}) satisfaz parcialmente Q_P . Conforme as etapas apresentadas anteriormente, o processamento da consulta em S_1 é realizado da seguinte forma: (a) $I_1 = S_{1A} \cap Q_J = S_{1A}$; (b) $Q_P \supset S_{1P}$; (c) $AQS_1 = \text{SELECT Hotel.nome FROM Hotel WHERE Hotel.diária} \geq 50 \text{ AND Hotel.diária} < 100 \text{ AND (Hotel.geometria IN } I_1)$; (d) $APQ \leftarrow APQ + AQS_1$; (e) $X \leftarrow X + I_1$; (f) $AS_1 = \text{SELECT Hotel.nome FROM } S_1$; e (g) $APQ \leftarrow APQ + AS_1$. Para o segmento S_1 , todo o seu conteúdo contribui com o resultado de Q , sendo que os dados restantes (i.e., referentes às diárias com preços entre \geq R\$ 50,00 e $<$ R\$ 100,00) são solicitados ao servidor pela *amending query* e armazenados em AQS_1 .

Por outro lado, analisando-se S_3 , verifica-se que o seu predicado (S_{3P}) satisfaz completamente Q_P . Neste caso, o processamento da consulta é da seguinte forma: (a) $I_3 = S_{3A} \cap Q_J$; (b) $Q_P \subseteq S_{3P}$; (c) $X \leftarrow X + I_3$; (d) $AS_3 = \text{SELECT Hotel.nome FROM } S_3 \text{ WHERE Hotel.diária} < 100 \text{ AND (Hotel.geometria IN } I_3)$; e (g) $APQ \leftarrow APQ + AS_3$.

Após o processamento da consulta em todos os segmentos candidatos, obtêm-se o vetor X e o resultado da consulta em cache APQ . A *reminder query* é executada no servidor da seguinte forma: “SELECT Hotel.nome FROM Hotel WHERE Hotel.diária < 100 AND ((Hotel.geometria IN Q_J) AND (Hotel.geometria NOT IN X))”.

5. REORGANIZAÇÃO DOS SEGMENTOS EM CACHE

Quando uma consulta é executada, se ela é totalmente respondida pelo servidor, toda a sua resposta é armazenada no cache semântico em um novo segmento. Entretanto, quando parte da resposta da consulta já está em cache, somente o resultado originário do servidor deve ser armazenado em um novo

segmento, uma vez que os segmentos do cache semântico devem possuir conteúdos disjuntos entre si (Ren and Dunham, 2000).

A forma na qual os segmentos semânticos são agrupados determina a granularidade dos segmentos e tem efeito significativo no desempenho do cache semântico (Lee et al., 1999; Ren et al., 2003). Existem várias alternativas para reorganizar os segmentos semânticos conforme as consultas são processadas. Para uma consulta Q e um segmento S , no modelo *partial coalescence in query* (Ren et al., 2003) cria-se um novo segmento com a resposta completa de Q e mantém-se S com o que restou ($S = S_p \wedge \neg Q_p$). Já no modelo *no coalescence* (Lee et al., 1999), após a utilização de S , são criados três segmentos disjuntos: (a) um segmento com a interseção de S e Q ($S_1 = S_p \cap Q_p$); (b) outro segmento com o que restou de Q ($S_2 = Q_p \wedge \neg S_p$); e (c) um terceiro segmento com o que restou de S ($S_3 = S_p \wedge \neg Q_p$).

A estratégia de reorganização de segmentos proposta neste artigo adota um modelo misto com relação aos modelos acima citados. Considerando que os predicados de uma consulta Q e de um segmento semântico S se relacionam, duas diferentes situações podem ocorrer (Figura 2). Na primeira situação (Figura 2a), todo o predicado de S é utilizado para responder Q . Como resultado, todo o conteúdo de S passa a ser relacionado ao novo segmento resultante da resposta completa de Q . Nesta situação, S é removido do cache semântico. Na segunda situação (Figura 2b), o predicado de S é utilizado parcialmente para responder Q . Assim, S continua existindo, porém passa a armazenar menos dados. A parte utilizada na resposta de Q é retirada de S e relacionada ao novo segmento resultante da resposta completa de Q .

Para realizar a reorganização dos segmentos em cache, além da análise do predicado, pode ser necessária a análise do relacionamento geográfico entre a consulta Q e um segmento semântico S . A situação mais complexa ocorre quando a área geográfica do segmento está relacionada parcialmente à janela de consulta e também o predicado do segmento contribui parcialmente com a resposta da consulta.

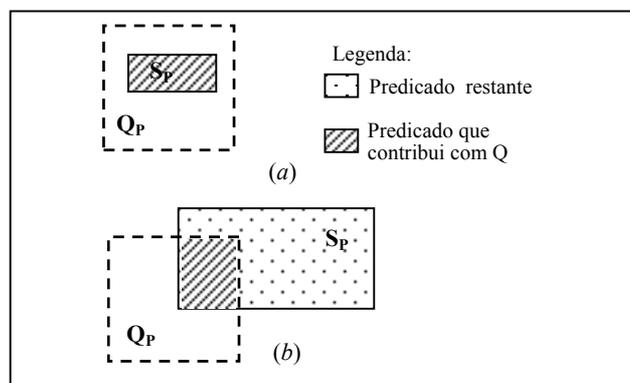


Figura 2. Relacionamento entre os predicados de Q e S.

No exemplo da Figura 3, a área de um segmento semântico S_1 intersecta a janela de consulta de Q (i.e., $S_{1A} \cap Q_J \neq \emptyset$) e o seu conteúdo contribui parcialmente com a resposta de Q (i.e., $S_{1P} \cap Q_P \neq \emptyset$). Assim, a reorganização de S_1 engloba os passos descritos a seguir. Esses passos são ilustrados na Figura 4.

- Retirar de S_1 o conteúdo que contribui com a resposta de Q (i.e., $S_{1P} \cap Q_P$) e atribuir esse conteúdo ao novo segmento S' , o qual será formado pela resposta completa de Q e terá área Q_J ;
- Criar um novo segmento S'' com área $I = S_{1A} \cap Q_J$, caso reste informação na área geográfica de interseção I, e armazenar esse conteúdo (i.e., $S_{1P} - (S_{1P} \cap Q_P) \neq \emptyset$ de I) em S'' . Essa etapa é necessária quando, por exemplo, S_1 armazena hotéis cujos valores de diária são menores do que R\$ 200,00 e Q solicita hotéis com valores de diária menores do que R\$ 100,00. Neste exemplo, hotéis com valores de diária maiores ou igual a R\$ 100,00 e menores do que R\$ 200,00 são armazenados em S'' ;
- Reduzir de S_1 a área de interseção I, ou seja, $S_{1A} \leftarrow S_{1A} - I$;
- Dividir a área geográfica de S_1 em sub-segmentos $S_{1A.1}, S_{1A.2}, \dots, S_{1A.n}$ caso a representação geométrica da área restante de S_1 seja complexa; e
- Excluir qualquer segmento ou sub-segmento que estiver vazio. Por exemplo, na Figura 4, se $S_{1A.1}$ não representar nenhuma informação, ele será excluído.

Após a reorganização de cada segmento, é necessário verificar se o CSDL possui espaço livre suficiente para acomodar os novos segmentos. Caso necessário, deve-se aplicar uma política de substituição de segmentos semânticos. Em Manica et al. (2004), é descrita a política ASCR, a qual complementa a proposta apresentada neste artigo.

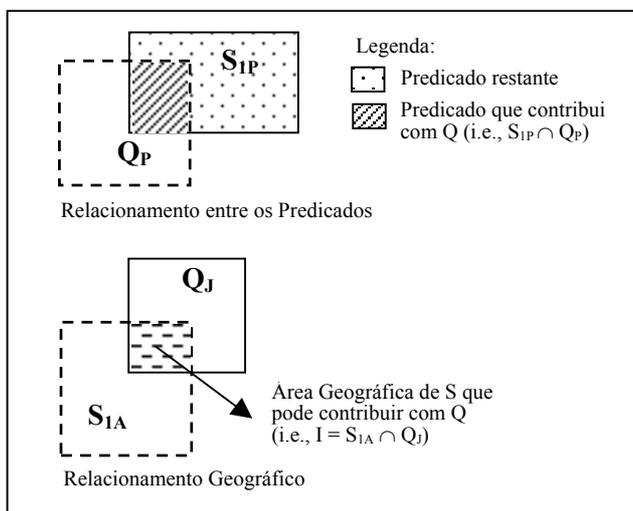


Figura 3. Relacionamento entre os predicados e relacionamento geográfico.

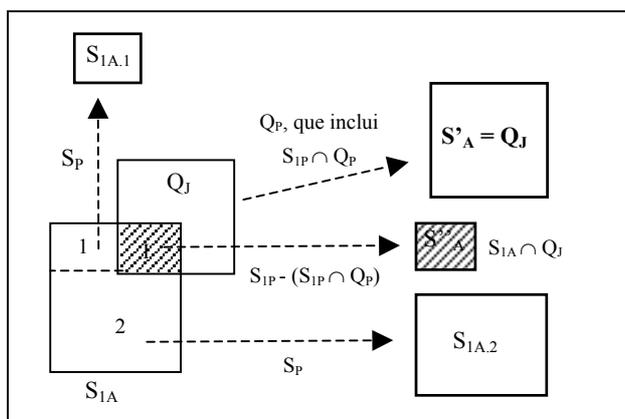


Figura 4. Reorganização do segmento S_1 .

6. CONCLUSÃO

Neste artigo, nós propusemos uma forma para realizar o processamento de consultas espaciais baseado no uso de um CSDL. Adicionalmente, nós discutimos sobre a reorganização de segmentos semânticos devido à execução de consultas espaciais.

A utilização de um CSDL possibilita que uma unidade móvel obtenha um melhor desempenho no processamento de consultas espaciais. Nesse sentido, o trabalho sendo desenvolvido na tese de doutorado da primeira autora representa uma possível solução.

As próximas etapas deste trabalho incluem:

- A implementação do modelo proposto para CSDL e dos algoritmos usados para gerenciá-lo;
- A realização de testes de desempenho que ilustrem o ganho obtido com o modelo proposto; e
- O refinamento da proposta de reorganização de segmentos de forma a distribuir de maneira uniforme a área gerada para os sub-segmentos.

REFERÊNCIAS BIBLIOGRÁFICAS

- Ciferri, R. R., 2002, Análise da Influência do Fator Distribuição Espacial dos Dados no Desempenho de Métodos de Acesso Multidimensionais, Tese de Doutorado, UFPE, Recife, Brasil.
- Ciferri, R. R., and Salgado, A. C., 2000, Performance Evaluation of Multidimensional Access Methods, in: *Proceedings of the 8th ACM Symposium on Advances in Geographic Information Systems*, Washington D.C, USA, pp. 183-184.
- Ciferri, R. R., Salgado, A. C., Times, V. C., Nascimento, M. A., and Magalhães, G. C., 2003, A Performance Comparison among the Traditional R-trees, the Hilbert R-tree and the SR-tree, in: *Proceedings of the 23rd International Conference of the Chilean Computer Science Society*, Chillán, Chile, pp. 1-15.
- Dar, S., Franklin, M. J., Jónsson, B. P., Srivastava, D., and Tan, M., 1996, Semantic Data Caching and Replacement, in: *Proceedings of the 22th International Conference on Very Large Data Bases*, Mumbai (Bombay), India, pp. 330-341.
- Dunham, M. H., and Kumar, V., 1998, Location Dependent Data and its Management in Mobile Databases, in: *Proceedings of the 9th International Workshop on Database and Expert Systems Applications*, Vienna, Austria, pp. 414-419.
- Gaede, V., and Günther, O., 1998, Multidimensional Access Methods, *ACM Computing Surveys*. **30**(2):170-231.
- Keller, A. M., and Basu, J., 1996, A Predicate-Based Caching Scheme for Client-Server Database Architectures, *VLDB Journal*. **5**(2):35-47.
- Lee, D. L., Xu, J., Zheng, B., and Lee, W., 2002, Data Management in Location-Dependent Information Services, *IEEE Pervasive Computing*. **1**(3):65-72.
- Lee, K. C. K., Leong, H. V., and Si, A., 1999, Semantic Query Caching in a Mobile Environment, *ACM Mobile Computing and Communications Review*. **3**(2):28-36.
- Manica, H., and Camargo, M. S., 2004, Caching Strategies for Mobile Computing Systems, in: *Proceedings of the 6th International Conference on Enterprise Information Systems*, Porto, Portugal, pp. 545-551.
- Manica, H., Camargo, M. S., Ciferri, R. R., and Ciferri, C. D. A., 2004, A New Model for Location-Dependent Semantic Cache Based on Pre-Defined Regions, in: *Proceedings of the 30th Conferencia Latinoamericana de Informática*, Arequipa, Peru, pp. 533-542.
- Ren, Q., 2000, Semantic Caching in Mobile Computing, PhD thesis, Southern Methodist University, USA.

- Ren, Q., and Dunham, M. H., 1999, Using Clustering for Effective Management of a Semantic Cache in Mobile Computing, in: *Proceedings of the 1st ACM International Workshop on Data Engineering for Wireless and Mobile Access*, Seattle, Washington, USA, pp. 94-101.
- Ren, Q., and Dunham, M. H., 2000, Using Semantic Caching to Manage Location Dependent Data in Mobile Computing, in: *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking*, Boston, MA, USA, pp. 210-221.
- Ren, Q., Dunham, M. H., and Kumar, V., 2003, Semantic Caching and Query Processing, *IEEE Transactions on Knowledge and Data Engineering*. **15**(1):192-210.
- Xu, J., Tang, X., and Lee, D. L., 2003, Performance Analysis of Location-Dependent Cache Invalidation Schemes for Mobile Environments, *IEEE Transactions on Knowledge and Data Engineering*. **15**(2):474-488.
- Xu, J., Tang, X., Lee, D. L., and Hu, Q., 1999, Cache Coherency in Location-Dependent Information Services for Mobile Environment, in: *Proceedings of the 1st International Conference on Mobile Data Access*, Hong Kong, pp. 182-193. Lecture Notes in Computer Science, volume 1748.
- Zheng, B., Xu, J., and Lee, D. L., 2002, Cache Invalidation and Replacement Strategies for Location-Dependent Data in Mobile Environments, *IEEE Transactions on Computers*, Special Issue on Database Management and Mobile Computing. **51**(10):1141-1153.