

Processamento Distribuído da Junção Espacial

MILTON RAMOS RAMIREZ

JANO MOREIRA DE SOUZA

COPPE/Sistemas-UFRJ, Centro de Tecnologia, H-319, CxP. 68.511. CEP 21.945-970
Instituto de Matemática-UFRJ, Centro de Tecnologia, C-114, CxP: 68.530. CEP 21.945-970
Cidade Universitária, Rio de Janeiro, RJ, Brasil.

milton@cos.ufrj.br

jano@cos.ufrj.br

Abstract. This paper describes a new strategy for distributed processing of intersect spatial join which is appropriate to the development of DSDBMS – Distributed Spatial Database Management Systems. We describe the experiments we carried out to compare the performances of our strategy to the others found in the literature. We have concluded that ours is suitable to DSDBMS and also more efficient than the other strategies.

1 Introdução

Atualmente são comuns Sistemas de Informações Geográficas – SIG – que utilizam a tecnologia dos sistemas distribuídos em rede de computadores [12][7]. Isso tem representado um desafio atual para as várias disciplinas envolvidas com o desenvolvimento de SIG.

Os SIGs baseados em sistemas distribuído – SIGD – podem ser definidos como sendo aqueles que possuem um variado número de sistemas geográficos autônomos, não necessariamente homogêneos, interconectados por uma rede de computadores e que cooperam na realização de *geoprocessamentos* com as bases de dados locais.

O componente de banco de dados de um SIGD deve ser capaz de gerenciar uma coleção de múltiplas bases de dados espaciais logicamente inter-relacionadas e distribuídas, oferecendo, pelo menos, a transparência da distribuição. Tal tipo de SGBD foi denominado como *SGBD Espacial e Distribuído* – SGBDED [1].

O desenvolvimento de SGBDED encerra diversas questões interessantes. Dentre elas consideramos o processamento distribuído e eficiente de consultas como sendo a principal. Isso porque ela é tanto uma questão central para qualquer tipo de SGBD, como também a eficiência de um SIG está ligada à eficiência do processamento de consultas do seu componente SGBD [6].

A discussão do processamento de consultas em SGBDED envolve vários aspectos. São relevantes o tipo de fragmentação e distribuição da base de dados espaciais, o tratamento de heterogeneidades, a autonomia dos sistemas locais que compõem o SGBDED, a arquitetura de distribuição de controles e os mecanismos de interoperação de dados, metadados e recursos [12][15].

Nesse âmbito podemos identificar questões específicas no processamento distribuído de consultas, tais como a localização de dados, fragmentação da consulta global segundo regras espaciais, geração e otimização do plano de execução e as questões relativas às estratégias para o processamento distribuído e eficiente de operações espaciais, tais como a junção espacial [12]. A pesquisa que desenvolvemos tem por foco essa última questão.

Assim, o objetivo deste trabalho é apresentar os nossos últimos resultados relativos à proposição de uma nova estratégia para o processamento distribuído da junção espacial com predicado de interseção entre dois conjuntos de dados espaciais que se encontram distribuídos entre dois componentes de um SGBDED.

Nossa estratégia foi desenvolvida tomando por base duas *diretivas* obtidas de nossas investigações sobre esse tipo de processamento distribuído. Em resumo, elas ditam que a estratégia para esse tipo de processamento deve aproveitar ao máximo o paralelismo inerente à distribuição e utilizar a interoperação para otimizar os processamentos locais e favorecer o paralelismo [12]. Esse fundamento tornou a nossa estratégia mais eficiente que outras encontradas até então na literatura. Tal fato foi comprovado através de resultados experimentais.

Este trabalho divide-se em mais seis seções. Na próxima, discutimos, no geral, as questões envolvidas com o processamento distribuído de consultas em SGBDEDs, focando no processamento distribuídos de operações espaciais, para o qual apresentamos as nossas diretivas. Na seção 3 discutimos as estratégias que encontramos na literatura e apontamos as suas limitações em relação as nossas diretivas. Na seção 4 apresentamos a nossa proposta de estratégia e, nas duas últimas seções, apresentamos os experimentos e resultados da comparação de desempenhos entre as estratégias.

2 Processamento Distribuído de Operações Espaciais

Em SGBD Espaciais e Distribuídos a seqüência de etapas do processamento de consultas não difere daquela de qualquer SGBD Distribuído. Entretanto, apesar dos objetivos de cada etapa permanecerem os mesmos, elas envolvem as complexidades identificadas para qualquer SGBD Espacial centralizado, o que torna esse processamento *não convencional* [13]. Assim, não se pode adotar diretamente as soluções estabelecidas para os SGBDs relacionais e distribuídos para resolver o processamento de consultas em SGBDED, uma vez que o processamento de consultas espaciais pode requerer transações longas e recursivas envolvendo dados que podem ser complexos e extensos.

Em resumo, as etapas do processamento de consultas em SGBDED encerram questões que são uma união entre aquelas levantadas para o processamento de consultas em SGBD distribuídos convencionais com aquelas identificadas para o processamento de consultas espaciais. A maioria delas ainda não se encontra totalmente estudada para todos os possíveis cenários de distribuição [15].

Encontramos na literatura alguns trabalhos sobre o processamento de consultas em SGBDED. Em [15] é proposto um protótipo de SGBDED para SIG Federados, implementado utilizando CORBA, onde são apresentadas contribuições à questão da geração de planos otimizados de execução da consulta. KUMAR et al. [7] contribuem com um protótipo de SIG orientado a objetos distribuído onde o processamento de consultas é implementado através de um módulo global que coordena vários outros módulos que gerenciam as bases componentes.

Trabalhos mais específicos são [1], [2], [12], [13], e [14]. Eles apresentam estratégias para o processamento distribuído da operação de junção espacial com predicado de interseção. AREF [2] propõe também uma estratégia distribuída para a seleção de pares de objetos mais próximos. O trabalho de TAN et al. [14] é um prosseguimento, pelos mesmos autores, do trabalho apresentado em ABEL et al. [1]. Ambos realizam experimentos com suas propostas utilizando um protótipo de SGBDED Federados. Iremos comentar essa estratégia em particular na seção seguinte.

RAMIREZ [12] discute todos os aspectos envolvidos com o processamento de consultas em SGBDED, abordando desde as questões gerais envolvidas com o desenvolvimento dos SGBDEDs até a questão específica do processamento distribuído de operações espaciais, discutindo a fragmentação de bases de dados espaciais, arquitetura do SGBDED segundo o grau de autonomia e problemas com objetos espaciais que transpassam fronteiras de subdivisões espaciais.

Não relevando todo o contexto de complicações envolvidas, nosso interesse neste trabalho está centrado na questão relativa às estratégias empregadas para o processamento distribuído da operação de junção espacial. RAMIREZ [12] aponta que há diversas alternativas de estratégias para esse tipo de questão, uma vez que as estratégias podem variar conforme o predicado da junção, tipos de fragmentação e distribuição da base de dados espaciais, grau de autonomia local dos SGBDEs componentes e existência de índices espaciais locais aos conjuntos de dados envolvidos.

No geral, as estratégias propostas na literatura estão baseadas na idéia de utilizar aproximações espaciais dos objetos para minimizar o custo de transmissão pela redução dos conjuntos de dados a serem transmitidos para a junção final. Todavia [1], [2] e [14] não aproveitam o paralelismo inerente à distribuição no processamento final da operação de junção, tampouco otimizam os processamentos geométricos locais, pois não dispõem de meios, sem custo adicional, para empregar algoritmos de reconhecida eficiência.

RAMIREZ et al. [13] observam, entretanto, que é um bom caminho a ser seguido a utilização de algoritmos como o proposto por BRINKHOFF et al. [5], utilizando índices espaciais locais e aproximações acuradas dos objetos espaciais, pois permitem, a um custo otimizado, a exclusão de objetos espaciais que não fazem parte da resposta final, permitindo assim diminuir os custos dos processamentos locais e minimizar o tráfego na rede de comunicação.

RAMIREZ [12] observou que haviam dois aspectos que podiam ser melhorados: o paralelismo da estratégia e um melhor aproveitamento dos índices espaciais locais por elas. A hipótese então era que se esses aspectos fossem contemplados o processamento distribuído de operações espaciais seria mais eficiente, algo comprovado pelo autor.

A razão disso tomou por base a meta de minimizar o custo total do processamento distribuído de operações espaciais em SGBDED, o que podia ser traduzido pelo objetivo de minimizar o volume de dados transferidos pela rede de comunicação, maximizar o processamento paralelo entre os SGBDE envolvidos, dividindo a carga entre eles, e minimizar a manipulação com objetos que não farão parte do resultado final da junção. RAMIREZ [12] então propõe duas diretrizes que devem ser seguidas na proposição de estratégias para o processamento distribuído de operações espaciais em SGBDED. São elas:

1. *Aproveitar ao máximo o paralelismo inerente ao processamento distribuído;*
2. *Utilizar o princípio da interoperação, no limite que a autonomia da arquitetura do SGBDED permitir, a fim de minimizar os processamentos locais.*

A primeira diretiva implica que a estratégia deve realizar a maior parte possível do processamento em paralelo, entre, pelo menos, os SGBDEs locais que contém os conjuntos de dados envolvidos na resolução da consulta. A segunda implica que a estratégia deve usar todas as possibilidades de compartilhamento, interdependência e interoperação de dados, metadados e recursos entre os SGBDEs componentes para otimizar os processamentos locais e favorecer o paralelismo.

3 Estratégias para o Processamento Distribuído da Junção Espacial de Interseção

O processamento da junção espacial de interseção entre dois conjuntos de dados espaciais pode ter custo elevado devido ao tamanho e complexidade dos objetos espaciais envolvidos [11]. Essa operação vem sendo tipicamente resolvida através de estratégias de filtragem e refinamento, utilizando aproximações espaciais dos objetos para reduzir o conjunto de candidatos a serem geometricamente processados [9]. A literatura é numerosa neste tema, havendo proposição de estratégias para o processamento da junção espacial em diversos cenários como o paralelo e para conjuntos de dados sem índices espaciais.

Assumiremos no restante deste trabalho o uso da estratégia de múltiplos passos - MSP - de BRINKHOFF et al. [5], onde os conjuntos de dados possuem índices, no caso *R*-Trees* [4]. Também, consideraremos que os conjuntos de dados estão distribuídos entre dois distintos nós - SGBD Espacial que compõem o SGBDED - ou seja, cada um dos conjuntos de dados encontra-se localizado em um computador diferente da rede. Esse tipo de cenário de distribuição é o mais assumido na literatura. Somente em [12] são apontados a existência de outros, em particular aqueles baseados em subdivisão espacial.

O processamento distribuído da junção espacial não pode ser simplesmente executado pela distribuição da operação de junção para os dois nós, os quais a executariam utilizando uma estratégia centralizada, para, em seguida, recolher e combinar os seus dois resultados. Isso porque qualquer dos dois nós não possui um dos conjuntos de dados, pois ele está localizado no outro nó, não permitindo a utilização direta do MSP.

Assim, o primeiro passo da MSP, relativa a *Junção-MBR*, não pode ser realizado pois ambos nós possuem somente *um* dos índices espaciais, enquanto que o outro índice reside remotamente no outro nó. Por motivo semelhante, o passo referente ao *filtro geométrico* utilizando aproximações acuradas dos objetos também não pode ser realizado pois cada nó possui somente um dos conjuntos de aproximações acuradas dos objetos. Entretanto, o último passo da estratégia poderia ser realizado se um dos nós recebesse o conjunto de dados

espaciais do outro nó envolvido. Todavia, esse caminho anularia por completo o esquema da estratégia MSP [5].

A idéia de transferir a totalidade de um dos conjuntos de dados para o outro nó é denominada na literatura como *estratégia ingênua* [1]. Uma estratégia ingênua que recuperasse os ganhos da MSP para processar a junção espacial no cenário distribuído citado poderia ser definida como a seguir, onde a Figura 1 apresenta um diagrama dos seus passos.

Estratégia Ingênua: Sejam dois conjuntos de dados R e S, localizados em dois SGBDEs componentes distintos, nó_R e nó_S respectivamente. Primeiro transfere-se a totalidade do *menor* dos conjuntos para o outro nó e, neste, constrói-se um índice *R*-Tree*. Em seguida executa-se a junção pelo algoritmo de múltiplos passos.

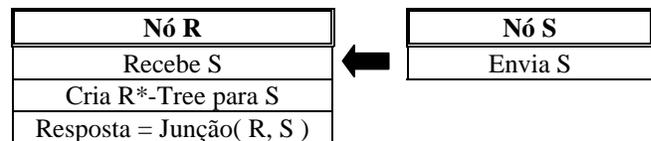


Figura 1 : Estratégia Ingênua

Essa estratégia Ingênua é aplicável a SGBDED com qualquer grau de autonomia, seja ele Integrado, Federado ou *Multidatabase* espacial [12].

Essa estratégia apresenta duas limitações. Primeiro, a operação de criação do novo índice pode ter um custo elevado dependendo da cardinalidade e do tamanho médio dos objetos do conjunto S. Alternativa para minimizar esse custo seria o uso de algoritmos de junção sem índices espaciais, tomando propostas tais como [11]. Segundo, mesmo que seja utilizada a melhor estratégia para a junção sem índice inicial, poderia haver custos desnecessários quando o número de objetos que atendem ao predicado da junção for pequeno em relação à cardinalidade dos conjuntos envolvidos. Esse custo desnecessário deve-se às operações de E/S e transmissões pela rede de comunicação de grande quantidade de objetos que não serão processados ao final e, também, deve-se à montagem de índice espacial com esses objetos desnecessários. Por fim, essa estratégia Ingênua tem grau de paralelismo zero, pois, como pode ser observado na Figura 1, em cada passo somente *um* dos SGBDEs componentes está trabalhando.

Todavia, há situações onde a resposta da junção espacial contém quase todos os objetos dos conjuntos envolvidos. Por exemplo, quando a junção espacial é utilizada em análises espaciais para calcular a sobreposição de dois mapas (*map overlay*), onde a maioria dos objetos se intercepta. Nesta situação os problemas citados são minimizados, em particular se a rede de comunicação tiver

uma taxa de transferência de dados elevada. Entretanto, à ausência de paralelismo persiste.

A literatura apresenta uma outra estratégia mais eficiente que a Ingênua, a qual é uma adaptação da estratégia para o processamento da junção relacional utilizada em SGBD convencionais distribuídos, onde a operação relacional *semijoin* é utilizada para reduzir o custo de transmissão [10].

Essa estratégia é a base da proposta espacial feita por ABEL et al. [1], a qual é estendida, pelos mesmos autores, para uso de *R-Trees* em [14]. Em [1] são resolvidas as questões relativas às dificuldades com a projeção do atributo espacial pela proposição de uma versão espacial do operador *semijoin*, o *spatial semijoin*. Esse operador pode ser definido como a seguir: sejam R e S dois conjuntos de dados espaciais e S' um conjunto de *aproximações* espaciais *conservativas* [5] dos objetos de S. O operador *spatial semijoin* entre R e S', para o caso do predicado de interseção, produz um subconjunto de todos os objetos de R tais que existe alguma aproximação de S' que o intercepte.

Essa definição implica que o resultado do *spatial semijoin* entre R e S' contém com certeza todos os pares que fazem parte do resultado final da junção. Porém pode conter mais alguns desnecessários, dependendo do tipo de aproximações utilizadas para gerar S' de S. Assim, a estratégia proposta em [1] foi definida como a seguir, onde a Figura 2 apresenta um diagrama dos seus passos:

Estratégia Semijoin: Sejam dois conjuntos de dados R e S, localizados em dois SGBDEs componentes distintos, nó_R e nó_S respectivamente, onde cada conjunto possui um índice *R-Tree* local:

1. No nó_S, obter S' como sendo o conjunto de aproximações espaciais conservativas e *distintas* dos objetos de S e o transmite para o nó_R;
2. No nó_R, calcular R' como sendo o *spatial semijoin* entre R e S' e enviá-lo para o nó_S;
3. No nó_S, realizar a junção espacial entre R' e S para obter o resultado final.

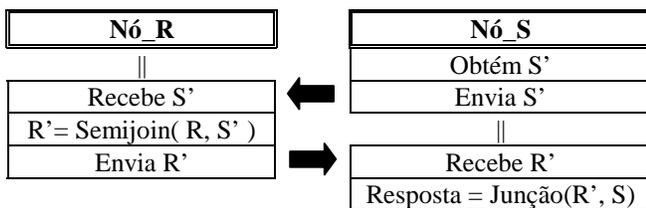


Figura 2 : Estratégia Semijoin

Analisando essa estratégia, observamos que ela reduz basicamente o custo de transmissão na rede, o qual era a

objetivo da estratégia original nos SGBD relacionais distribuídos, uma vez que ela supõe que o custo de transmissão dos subconjuntos R' e S' é menor que o da transmissão total do menor entre os conjuntos R e S. As implementações dessa estratégia, apresentadas em [1] e [14], demonstram também que ela induz uma redução do custo de CPU e de E/S, uma vez que reduz a manipulação com objetos que não contribuirão para o resultado final.

Apesar da estratégia Semijoin ser mais eficiente que a Ingênua, com algumas exceções [14], podemos observar que há ainda o que melhorar. O problema mais evidente é o fato dessa estratégia, como a Ingênua, também *não* aproveitar o *paralelismo* inerente ao processamento distribuído. Observando a Figura 2 podemos constatar que a cada passo somente *um* dos SGBDEs locais está rodando, ou seja, é um processamento sequencial dividido entre dois computadores. Desse modo, as duas estratégias possuem o mesmo custo de CPU para os processamentos geométricos. A razão disso é a origem da estratégia Semijoin, a qual tinha por objetivo apenas minimizar o tráfego na rede de intercomunicação, uma vez que em SGBD relacionais distribuídos o tempo de processamento local é irrelevante em relação ao de comunicação, suposição que não é válida para o contexto espacial [6][14].

Também, como pode ser confirmado pelos experimentos de ambos os trabalhos, não há índices espaciais para os subconjuntos S' e R' recebidos. Assim, não se pode aplicar diretamente, sem custos adicionais, uma estratégia eficiente para os processamentos das junções locais. Por fim, na transmissão de R' como um todo para o nó_S podem ocorrer situações onde é alto o custo de transmissão, algo confirmado pelos resultados que indicam que o melhor caso para essa estratégia ocorre quando os objetos de R e S possuem tamanhos médios próximos [14], isso porque quando os objetos de S forem muito menores do que os de R, o mais vantajoso seria transmitir os pares de S correspondentes aos objetos em R'.

Outras situações onde a estratégia Semijoin apresenta dificuldade, com resultados próximos ou piores que os da estratégia Ingênua, ocorrem quando o *spatial semijoin* não produz uma boa redução do subconjunto R, quer seja por aceitar muitos objetos que não atendem ao predicado da junção, quer seja pelo fato de quase todos os objetos possuírem interseção com algum outro objeto [14].

Em resumo, essas estratégias apresentadas não realizam em paralelo os processamentos geométricos, os quais podem ser de custo elevado, como também não utilizam algoritmos eficientes para a junção espacial, como proposto na literatura, pois não resolvem o problema questão quanto à não disponibilidade dos índices locais e aproximações acuradas do outro conjunto de dados remoto.

4 Estratégia MR2

Em vista das dificuldades observadas nas duas primeiras estratégias abordadas, nessa seção propomos uma nova estratégia para o processamento distribuído da junção espacial de interseção. Desenvolvemos nossa estratégia segundo as duas diretivas apresentadas na seção 2. Baseados na primeira diretiva, tornamos paralela a etapa do processamento geométrico local, pois o seu custo é considerado dominante no processamento de consultas em SGBD Espaciais [6]. A segunda diretiva foi aplicada para resolver, através de compartilhamento, o problema relativo ao não aproveitamento de índices espaciais remotos, permitindo a otimização dos processamentos locais.

Assim, em nossa proposta consideramos que o custo do terceiro passo da MSP, relativo ao processamento com a geometria exata, domina o tempo total de processamento, em particular quando a consulta requer o cálculo da interseção entre os objetos. Em conjunto, a nossa estratégia preconiza o aproveitamento de qualquer índice espacial em processamentos locais remotos, ou seja, propomos que os índices espaciais possam ser *compartilhados*. A idéia central é permitir que qualquer índice local ou aproximações acuradas dos objetos possam ser utilizados remotamente por qualquer nó para os seus processamentos locais. Isso permite eliminar o custo da construção do índice para conjuntos de dados recebidos e empregar diretamente a estratégia MSP [5] para realizar junções locais que envolvam conjuntos de dados remotos. Para que isso seja possível os índices espaciais devem ser alterados de tal forma a serem *interoperáveis*. Para o caso de uma *R*-Tree*, propomos a modificação do tipo de referências relativas utilizadas nos nós da *R-Tree* de tal modo que elas sejam referências *globais* ao SGBDED. Desse modo, deverão ser alterados os algoritmos relativos à *R*-Tree* para que reflitam o uso de referências globais.

Por outro lado, buscando minimizar o custo de E/S, a literatura sobre junção baseada em MSP propõe que as folhas da *R*-Tree* contenham as aproximações acuradas dos objetos a serem utilizadas no passo do filtro geométrico. Isso é razoável quando o processamento é centralizado pois otimiza E/S [5]. Entretanto, para o contexto da nossa estratégia distribuída, isso levaria à transmissão das aproximações acuradas quando da transmissão da *R*-Tree*. Como as aproximações acuradas dos objetos, em geral, não são tão pequenas como MBRs (*Minimum Bounding Rectangle*) [16], isso pode significar um custo adicional desnecessário, em particular quando a cardinalidade do conjunto de dados é grande e o número de objetos que atendem ao predicado da junção é pequeno. Nesse caso, haverá a transmissão de muitas aproximações que não serão utilizadas pelo filtro geométrico, aumentando desnecessariamente o custo de E/S e de comunicação.

Uma solução simples, porém eficaz para essa situação, seria transmitir somente um subconjunto das aproximações relativas aos objetos do conjunto de pares candidatos produzido pela primeira etapa de *Junção-MBR*. Para tanto propomos duas outras alterações. Primeiro, as *R*-Trees* não mais conteriam em suas folhas as aproximações acuradas dos objetos, mas somente os seus respectivos MBRs e uma referência *global* para as respectivas aproximações acuradas, as quais seriam armazenadas em um conjunto separado. Desse modo seria apenas transmitida a *R*-Tree* sem as aproximações acuradas as quais seriam solicitadas pelas etapas seguintes.

Assim, a nossa estratégia que utiliza índices espaciais compartilhados e processamento paralelo pode ser definida como a seguir e ilustrada na

Figura 3. A denominamos MR2 – *Múltiplos passos com índices Remotos*, versão 2 (a versão 1 seria com *R-Trees* contendo as aproximações [12])

Estratégia MR2: Sejam dois conjuntos de dados M e E, localizados em dois SGBDEs componentes distintos, nó_M e nó_E. Cada nó possui uma *R*-Tree* local, *rt_M* e *rt_E* respectivamente, contendo aproximações MBR dos seus objetos e um conjunto separado contendo aproximação mais acurada dos seus objetos para o passo do filtro geométrico. Os passos da estratégia são os seguintes:

1. Identificar o menor entre os índices espaciais *rt_M* e *rt_E*. Supondo que *rt_E* é menor que *rt_M* então o nó_M será o nó *Mestre* e o nó_E será o nó *Escravo*;
2. Transmitir o índice do nó *Escravo* para o nó *Mestre*;
3. No nó *Mestre* realizar a *Junção-MBR*, gerando um conjunto de pares candidatos ME1. Em seguida, requisitar ao nó *Escravo* as respectivas aproximações acuradas remotas relativas ao conjunto ME1;
4. No nó *Mestre*, executar o filtro geométrico, para reduzir os pares que não fazem parte do resultado final, gerando um novo conjunto de pares ME2, o qual contém tanto os pares que atendem ao predicado da junção como os pares que permanecem inconclusivos;
5. No nó *Mestre*, dividir o conjunto ME2, gerando dois conjuntos de pares ME2_M e ME2_E, de tal modo a balancear a carga para o processamento paralelo e minimizar o custo de transmissão. Transmitir para o nó *Escravo* o seu conjunto de pares de ME2_E;
6. Em paralelo, nó *Mestre* e *Escravo* requisitam os objetos remotos dos seus respectivos conjuntos de pares ME2_M e ME2_E;
7. Em paralelo, nó *Mestre* e *Escravo* executam o passo relativo à junção com a geometria exata dos objetos para os seus respectivos conjuntos ME2_M e ME2_E.

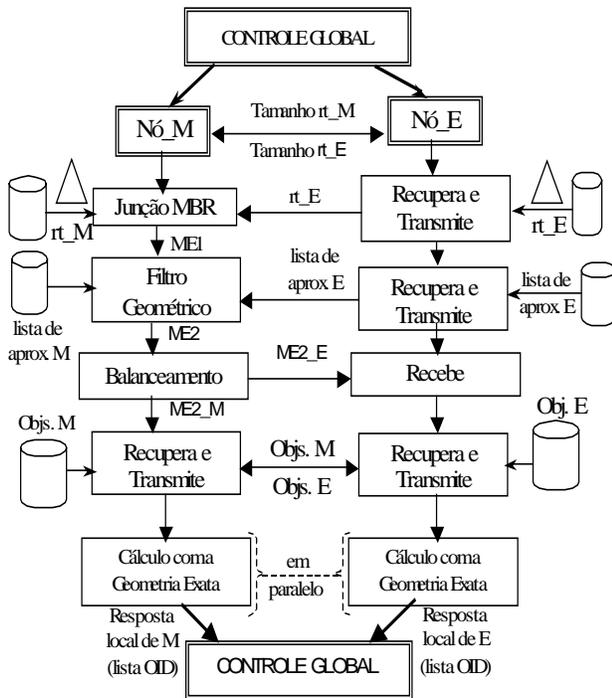


Figura 3 : Estratégia MR2

A estratégia MR2 procura evitar o processamento de objetos que não farão parte do resultado final, o que otimiza todos os custos envolvidos. Também, minimiza o custo da comunicação através da escolha dos menores dados a serem transmitidos. Por ser paralela, otimiza os processamentos geométricos e, por utilizar índices espaciais compartilhados, torna eficiente as junções espaciais locais, isso dispensando o custo da montagem de índice para conjuntos de dados remotos.

Tomando somente o fato dessa estratégia ser paralela, podemos constatar que ela é mais eficiente que as anteriores em contextos onde é dominante o custo de CPU, caso geral da junção espacial. Em particular, essa estratégia é mais eficiente quando quase todos os objetos atendem ao predicado da junção; quando é relevante a diferença entre os tamanhos médios dos objetos; quando o tamanho dos conjuntos de dados for grande, ou seja, uma combinação entre altas cardinalidades e objetos com tamanho médio grande; quando o custo de transmissão é dominante, ou seja, é baixa a taxa de transferência de dados na rede de comunicação, ou o tráfego de dados nela é alto, caso típico da Internet nos tempos atuais; e quando a seletividade das aproximações espaciais não é boa, isto é, os conjuntos ME1 e ME2 possuem muitos pares de objetos que não atendem ao predicado da junção [12].

O pior caso para a estratégia MR2 ocorre quando as cardinalidades dos conjuntos de dados são pequenas, em combinação com um número pequeno de pares no conjunto

ME2, ou seja, poucos objetos atendendo ao predicado da junção. Isso porque se torna pequeno o custo de CPU, devido à necessidade de poucos testes geométricos ou de processamentos com a geometria exata, levando à predominância do custo de comunicação, pois, a princípio, seria pequeno o custo de E/S, devido ao fato de serem poucos os objetos processados. Entretanto, mesmo nesse pior caso, a estratégia MR2 ainda apresentou desempenho ligeiramente melhor que os das outras duas estratégias, como foi demonstrado pelos experimentos, em particular devido ao balanceamento de carga e o compartilhamento de índices.

5 Experimentos

Os experimentos realizados objetivaram a comparação entre o desempenho das estratégias MR2, Ingênua e Semijoin. Assim, medimos tempo total gasto, tempo de E/S, das transmissões de dados pela rede e tempo do processamento geométrico. Também medimos o volume de transmissão de dados e de E/S.

Implementamos as estratégias no contexto específico do núcleo de processamento de um SGBDED integrado e homogêneo. Entretanto, como não dispúnhamos de um protótipo de SGBDED operacional, optamos por implementar uma *bancada de testes* para essa finalidade, cuja arquitetura pode ser encontrada em [12].

As estratégias foram implementadas como algoritmos distribuídos [3] a serem executados entre um Monitor de Execução Global *cliente* e vários Monitores de Execução Locais *servidores*. A bancada de testes foi implementada em C++ utilizando a biblioteca de troca de mensagens MPI para implementar a parte de intercomunicação. Utilizamos uma rede com oito computadores Pentium® III/450 MHz (64 Mbytes/NT Server 4.0). A rede possuía taxa de transferência entre 7 e 4 Mbits/s.

Os objetos persistentes foram implementados com uma identificação global, isto é, uma identificação única válida para todos os sistemas componentes do SGBDED. Assim, tanto os objetos espaciais, como os objetos que armazenavam índices espaciais e aproximações tinham um *OID global*. Isso permitiu a interoperação deles entre os diversos SGBDEs locais, em particular o compartilhamento de índices locais e aproximações acuradas dos objetos.

A base de dados espaciais utilizada foi produzida a partir do conjunto de mapas do macrodiagnóstico da zona costeira do Brasil produzidos em 1996 pelo Ministério do Meio Ambiente dos Recursos Hídricos e da Amazônia Legal. Os objetos gerados foram polígonos, as *R*-Trees* utilizaram aproximações MBR como chave espacial e as aproximações acuradas foram baseadas na assinatura *raster 4CRS* proposta por ZIMBRÃO et al. [16].

Os conjuntos de dados foram divididos em três grupos. O primeiro grupo era composto de conjuntos de objetos com mesmo número médio de vértices que variavam na cardinalidade de objetos (grupo CDD com 1.000 a 85.000 objetos). No segundo grupo os conjuntos de objetos diferiam no tamanho médio dos objetos espaciais (grupo TMOE com objetos de 110, 57 e 19 vértices). O último grupo combinava conjuntos que variavam a percentagem de objetos que atendiam ao predicado da junção (grupo CRCR com 10% a 90% dos objetos participando da resposta final).

6 Resultados

Os resultados obtidos dos vários experimentos que realizamos indicaram não somente que a estratégia MR2 é adequada para os SGBDED como também justificaram as afirmações do porque ela é mais eficiente que as demais.

A estratégia MR2 *sempre* teve um desempenho melhor que as demais em todas as situações testadas. Tomando o *tempo total* da consulta, a estratégia MR2 foi, no pior caso, 31% melhor que a estratégia Semijoin e 67% melhor que a Ingênuo, como a Figura 4 apresenta para um conjunto com 50K objetos de 19 vértices em média e 32% de objetos atendendo à junção. A explicação geral para esse fato é que a estratégia MR2 otimiza *todos* os aspectos envolvidos com o processamento da junção espacial de interseção.

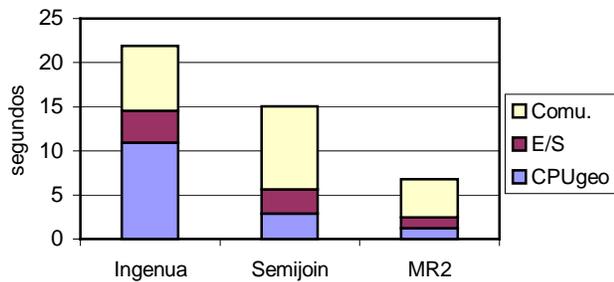


Figura 4 : Tempos acumulativos, CCD_G19_50K

O principal ganho está relacionado com o maior paralelismo da MR2, levando-a a superar a Semijoin mesmo quando é baixo o número de objetos que atendem ao predicado, ou a cardinalidade dos conjuntos e o tamanho médio dos objetos são grandes. Isso pode ser observado nas medidas de tempo de CPU como apresenta a Figura 5, onde o ganho variou de 58% a 49% do tempo da Semijoin e foi 82% da Ingênuo, para conjuntos com 1K a 20K objetos com 112 vértices em média e 30% de participação deles no resultado final. Também, o custo de CPU será tanto maior quanto maior for o número de objetos que atendem ao predicado, ou maior o tamanho médio dos objetos. Isso foi observado nos experimentos com o grupo CRCR, onde a

estratégia MR2 obteve, no pior caso, um ganho de 49% em relação à estratégia Semijoin e 87% em relação à Ingênuo.

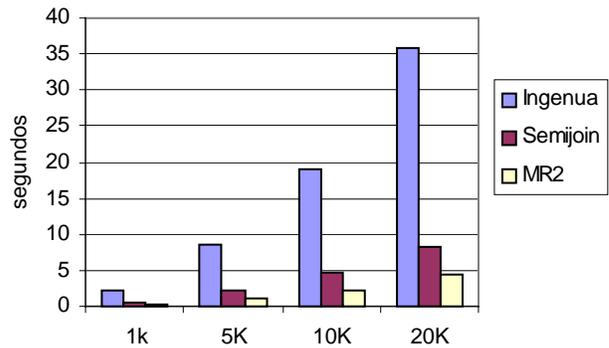


Figura 5 : Tempo de CPU para CCD_G112

Quanto ao fato da estratégia MR2 não possuir o custo de montagem do índice, pois compartilha índices remotos, esse ganho pode ser constatado quando a cardinalidade dos conjuntos de dados envolvidos foi grande e o tamanho médio dos objetos também.

Avaliando o custo de comunicação, a estratégia MR2 superou as demais pois minimiza todas as transferências de dados. Primeiro, igualmente a Semijoin, ela elimina com o uso de índices e filtragem geométrica a transferência de objetos que não farão parte do resultado final. Segundo, devido ao tipo de balanceamento, ela transfere os menores objetos selecionados para o passo final. Esse critério em particular fez com que ela superasse a Semijoin quando da variação do número de objetos que atendem ao predicado da junção, como apresenta a figura 6, onde o ganho variou de 75% a 1%. O pior caso ocorre quando quase todos os objetos atendem ao predicado e devem ser transferidos.

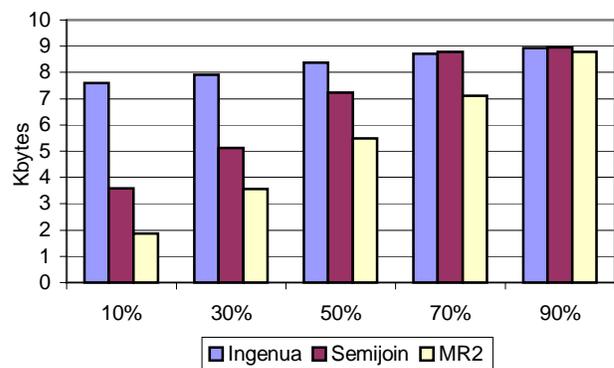


Figura 6 : Volume de comunicação para CRCR.

Avaliando somente o custo de comunicação e E/S, a estratégia MR2 supera as demais quando os conjuntos de dados são grandes e têm-se poucos objetos atendendo ao predicado da junção. No caso oposto, o volume de E/S da MR2 equipara-se ao da Semijoin, todavia ambos ficam

abaixo da Ingênuia pois ela realiza uma varredura adicional no conjunto de dados para a construção do índice, como pode ser visto na Figura 7 a seguir.

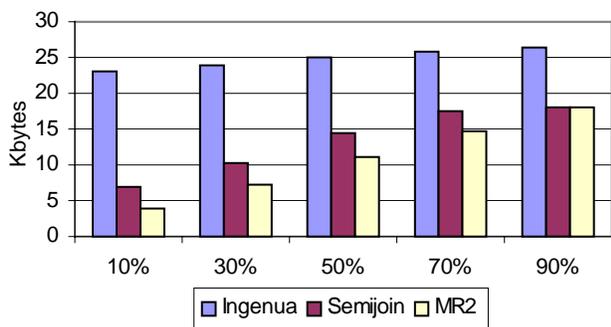


Figura 7 : Volume de E/S para CRCR

7 Conclusões

Nossos resultados reforçaram as evidências de que é relevante o aproveitamento máximo do paralelismo inerente ao processamento distribuído. Isso porque a execução de uma consulta espacial frequentemente envolve um processamento longo e recursivo, o qual manipula com dados complexos e extensos. Então, se não houver pelo menos o processamento paralelo entre os sistemas componentes que contêm os conjuntos de dados envolvidos, esse processamento ainda pode ser otimizado.

Nossos resultados indicaram que o atendimento à nossa diretiva quanto ao uso de interoperação, traduzida pelo compartilhamento de índices locais, pode permitir ao processamento distribuído de operações espaciais utilizar as estratégias centralizadas desenvolvidas para a junção com índices espaciais e emprego de filtros geométricos baseados em assinaturas, como proposto por [16].

Em resumo, quando tratamos de SGBD Espaciais e Distribuídos, as estratégias empregadas devem buscar a melhor interoperação permitida pela autonomia a fim de tornar os processamentos locais mais eficientes, como se o processamento fosse *centralizado* em cada SGBDE local. Também, as estratégias não devem negligenciar o potencial de paralelismo que pode ser estabelecido entre os sistemas envolvidos no processamento distribuído.

Neste trabalho fizemos uma série de restrições. Todavia, nossas investigações consideram outras possibilidades não abordadas as quais abrem um campo vasto de direções a serem tomadas em pesquisas futuras, em particular quando consideramos a investigação em cenários Federados e *Multidatabases*, com uso de XML e CORBA. Também estamos investigando outros tipos de operações e predicados, em particular a junção espacial de proximidade, isso considerando conjuntos sem índices

espaciais, ou com outros métodos de acessos espaciais que não sejam baseados em *R-Trees*, e, também, com conjuntos de dados lineares e temporais.

Referências Bibliográficas

- [1] Abel, D.J., Ooi, B.C., et al., "Spatial Join Strategies in Distributed Spatial DBMS". *Advances in Spatial Databases*, LNCS n. 951, Springer, pp.348-367, 1995
- [2] Aref, W., "Query Processing in Distributed Spatial Databases". *1st Inter. Conf. Interoperating GIS*, Santa Barbara, California, Dec. 1997.
- [3] Barbosa, V.C., *An Introduction to Distributed Algorithms*, Cambridge, MIT Press, 1996.
- [4] Beckmann, N., Kriegel, H-P, Schneider, R., et al., "The R*-Tree: An Efficient and Robust Access Method for Points and Rectangles". *Proceedings of ACM SIGMOD Conference*, pp. 322-331, 1990.
- [5] Brinkhoff T., Kriegel H.-P., Schneider R., et al., "Multi-Step Processing of Spatial Joins". *Proc. of ACM SIGMOD Conference*, pp. 197-208, 1994.
- [6] Güting, R.H., "An Introduction to Spatial Database Systems", *The VLDB Journal*, v.3,n.4,pp357-399,1994
- [7] Harder, C., *Serving Maps on the Internet: Geographic Information on The World Wide Web.*, ESRI, 1998.
- [8] Kumar, B.V., Sinha, P., Bhatt, P.C. P., "DoGIS: A distributed and object oriented GIS". *Proc. of VII Inter. Sym. Spatial Data Handling*, pp.263-275. 1997.
- [9] Orenstein, J.A., "Spatial Query Processing in an Object-Oriented Database System." *Proceedings of ACM SIGMOD Conference*, pp. 326-336, 1986.
- [10] Özsu, M.T., Valduriez, P., *Principles of Distributed Database Systems*. 2ed, Prentice-Hall,1999.
- [11] Patel, J.M., D.J. Dewitt, "Partition Based Spatial-Merge Join". *Proc.ACM SIGMOD*, pp.259-270, 1996.
- [12] Ramirez, M.R., *Processamento Distribuído de Consultas Espaciais*. Tese D.S.c, COPPE/UFRJ, 2001.
- [13] Ramirez, M.R., Souza, J.M., Mattoso, M.L.Q., *Operações Espaciais em Banco de Dados Espaciais Distribuídos*. RT-ES- 477/98, COPPE /UFRJ, 1998.
- [14] Tan, K-L., Ooi, B. C., Abel, D. J., "Exploiting Spatial Indexes for Semijoin-Based Join Processing in Distributed Spatial Databases", *IEEE Transaction on Knowledge and Data Engineering*, v. 12, n. 6, pp. 920-937, Nov 2000.
- [15] Wang, F., "A Distributed Geographic Information System on the CORBA". *GeoInformática*, v. 4, n. 1, pp. 89-115, Mar 2000.
- [16] Zimbrão, G.S., Souza, J.M., "A Raster Approximation For Processing of Spatial Joins". *Proc. of 24th Inter. Conf. VLDB*, pg 24-27, Aug. 1998.