FRAGMENTAÇÃO FÍSICA DE DADOS EM DATAWAREHOUSES BASEADA EM ÁRVORES R.

Rogea Rocha Silveira

DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE DA COORDENAÇÃO DOS PROGRAMAS DE PÓS-GRADUAÇÃO DE ENGENHARIA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS EM ENGENHARIA DE SISTEMAS E COMPUTAÇÃO.

Aprovada por:

Prof. Claudio Esperança, Ph. D.

Prof. Geraldo Zimbrão da Silva, D. Sc.

Profª. Vanessa de Paula Braganholo, D. Sc.

RIO DE JANEIRO, RJ, BRASIL
SETEMBRO DE 2007

SILVEIRA, ROGEA ROCHA
Fragmentação Física de Dados em Data Warehouses Baseada Em Árvores
R. [Rio de Janeiro] 2007
IX, 71p. 29,7 cm (COPPE/UFRJ, M.Sc., Engenharia de Sistemas e Computação, 2007)
Dissertação – Universidade Federal do Rio de Janeiro, COPPE
1. Fragmentação Física
2. Índices Multidimensionais
3. Paralelismo
I. COPPE/UFRJ II. Título (série)

Aos meus pais e ao meu esposo Rodrigo, por todo apoio.

AGRADECIMENTOS

Agradeço ao meu orientador, professor Geraldo Zimbrão por toda sua dedicação, seu incentivo e apoio que foram fundamentais para a conclusão dessa dissertação. Sua participação em minha carreira acadêmica, ocorrida em vários momentos: quando foi meu professor de disciplinas na graduação, no mestrado e orientador de projeto final de curso, foi muito importante na minha formação e servirá de exemplo em toda minha carreira.

Agradeço aos professores Cláudio Esperança e Vanessa Braganholo por terem aceitado o convite de participação na banca. Com certeza todas as sugestões contribuirão muito para nosso trabalho.

Agradeço ao professor Jano de Souza, chefe da linha de Banco de Dados, pelo apoio tanto na carreira acadêmica quanto profissional e por se dedicar em manter o nível de excelência da linha de pesquisa.

Agradeço ao professor Blaschek pelo apoio durante os anos de projeto COPPETEC, onde pude aplicar meus conhecimentos acadêmicos e desenvolver meu lado profissional.

Agradeço à Patrícia Leal, secretária de linha de banco de dados, e à Solange Santos, secretária acadêmica do PESC/UFRJ por estarem sempre dispostas a ajudar.

Agradeço à minha família por todo o carinho. Aos meus pais por privilegiarem minha educação e terem investido no meu futuro. Tudo que sou hoje devo a eles.

Agradeço ao meu esposo Rodrigo pelo apoio incondicional, pelas palavras de carinho nos momentos em que eu mais precisava e pela paciência e compreensão nos momentos em que não pude estar presente. Essa conquista também é sua. Obrigada por tudo!

Agradeço aos meus amigos, Vinicios Pereira, Amanda Varella, Vinicius Vonheld e Camille Furtado pelo incentivo e apoio em todos os momentos. A amizade de vocês também foi fundamental nessa jornada.

Resumo da Dissertação apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M. Sc.)

FRAGMENTAÇÃO FÍSICA DE DADOS EM DATA WAREHOUSES BASEADA EM ÁRVORES R.

Rogea Rocha Silveira

Setembro/2007

Orientador: Geraldo Zimbrão Da Silva

Programa: Engenharia de Sistemas e Computação

Data warehouses (DW) integram e acumulam grande volume de dados históricos provenientes das transações operacionais das organizações e de várias fontes externas para dar suporte ao processo de tomada de decisão das organizações. Consultas complexas, de alto custo e que exigem grande poder de processamento do SGBD são executadas sobre esse repositório para extração de informações estratégicas para a organização. Esta dissertação tem como objetivo melhorar o desempenho desse tipo de consulta levando-se em consideração a forma como a modelagem multidimensional organiza os dados de um DW. O esquema estrela gerado por essa modelagem é composto por tabelas dimensão e por tabelas fato e pode ser visto como um espaço multidimensional chamado de cubo de dados. Nós propomos uma abordagem que particiona esse espaço através de uma estrutura de indexação multidimensional, no nosso caso, a árvore R, utilizada para fragmentar fisicamente a tabela fato de um esquema estrela. A alocação dos fragmentos em um ambiente distribuído possibilita que as consultas sejam processadas através do paralelismo intra-consulta, que pode ajudar a reduzir consideravelmente o tempo de execução das consultas individualmente. Para validar nossa proposta, desenvolvemos um protótipo onde rodamos experimentos em um ambiente distribuído de 8 nós usando o *benchmark* TPC-H. Os resultados mostram que nossa proposta permite limitar o volume de dados a ser acessado por determinadas consultas, reduzindo desta forma seu tempo de execução.

Abstract of Dissertation presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

PHYSICAL PARTITIONING OF DATA WAREHOUSES BASED IN R-TREES

Rogea Rocha Silveira

September/2007

Advisor: Geraldo Zimbrão Da Silva

Department: Engenharia de Sistemas e Computação

Data warehouses (DW) integrate and accumulate large volumes of historic data from operational transactions of organizations and from various external sources, to support the organization's decision making process. Complex queries, costly and the one's that demand high processing power from the database engine, are run over the aforementioned repository to extract reports of strategic information for the organization. The objective of this work is to improve the performance of this type of queries taking into account the way in which multidimensional modeling organizes the data of a DW. The star scheme generated by this modeling is composed of dimension and fact tables and can be seen as a multidimensional space called data cube. Our proposed approach is to partition this space through a multidimensional indexing structure, in this case the R-tree that we use to physically fragment the fact table of a star scheme. The fragment allocation in a distributed environment allows queries to be processed through intra-query parallelism, which can help reduce considerably the execution time of queries individually. To validate our approach, we implemented a prototype where we ran experiments in an 8 nodes distributed environment using the TPC-H benchmark. The results have shown that our approach allows limiting the volume of data to be accessed by certain queries, thus reducing its execution time

ÍNDICE

LISTA DE FIGURAS

1. Introdução

Um *data warehouse* é um repositório de dados que integra e acumula dados históricos gerados por transações operacionais. Ele é especialmente modelado para apoiar o processo de tomada de decisão nos negócios de uma organização (INMON, 2005). Aplicações OLAP (*On-line analytical processing*) acessam esse repositório e disponibilizam consultas onde os analistas de negócio podem extrair diversas informações estratégicas sobre o negócio da organização. Essas consultas caracterizam-se por serem complexas, de alto custo e por acessarem grandes volumes de dados. Garantir eficiência no processamento desse tipo de consulta é fundamental para que os analistas de negócio tenham acesso às informações rapidamente e assim possam manter a organização competitiva frente à sua concorrência.

Entre as estratégias utilizadas para alcançar maior desempenho nesse cenário, destacam-se a utilização de processamento distribuído, onde ganho de desempenho é alcançado através de um bom projeto de distribuição de dados que inclui a fragmentação dos dados e a alocação desses fragmentos entre os nós do ambiente distribuído. Os sistemas de computação distribuída consistem em um conjunto de elementos de processamento independentes que estão interconectados por uma rede de computadores e que cooperam entre si na execução de suas tarefas atribuídas (ÖZSU e VALDURIEZ, 1999). O processamento distribuído se baseia na idéia de que é mais eficiente dividir um problema complexo em fragmentos menores, de resolução mais simples, que podem ser processados por diferentes dispositivos computacionais.

Em sua maioria, os sistemas distribuídos representam um método mais econômico para alcançar maior poder computacional. Um exemplo que vem ganhando cada vez mais destaque são os agrupamentos de computadores pessoais. Um agrupamento de computadores pessoais é um sistema local que consiste em um conjunto de computadores independentes interconectados em uma rede de alta velocidade (STERLING, 2001). Atualmente é considerada uma plataforma bem atraente economicamente para processamento paralelo e aceleração de transações, comparado aos servidores de arquiteturas paralelas com multiprocessadores fortemente acoplados, que além do hardware especializado, exigem também software especial, o que encarece muito essa opção (CECCHET, MARGUERITE et al., 2004). Além disso, os sistemas distribuídos são mais flexíveis, pois permitem um crescimento incremental através da

inclusão de novos dispositivos computacionais caso seja necessário alcançar maior desempenho.

O projeto de distribuição de dados consiste em determinar a maneira pela qual os dados são distribuídos pelos nós de uma rede (ÖZSU e VALDURIEZ, 1999). Ele é feito em duas etapas. Primeiramente, é definido o projeto de fragmentação da base, onde as tabelas são decompostas em partes menores, chamados de fragmentos. Em seguida, o projeto de alocação é responsável por definir como esses fragmentos serão alocados entre os nós da rede.

A distribuição dos dados permite que as consultas sejam processadas paralelamente. Vários trabalhos empregaram o processamento paralelo em diferentes cenários e alcançaram bons resultados ((LIMA, 2004), (FURTADO C., 2006), (MATTOSO, ZIMBRÃO, LIMA, 2005), (MYSQL, 2006), (SWAMINATHAN, 2005), (ORACLE, 2003)). O processamento paralelo pode ser feito de duas maneiras: através do paralelismo inter-consulta ou do paralelismo intra-consulta. O paralelismo inter-consulta consiste em executar paralelamente consultas distintas nos nós da rede. Seu objetivo é aumentar a vazão do sistema. Já o paralelismo intra-consulta consiste em subdividir uma mesma consulta em sub-consultas que são executadas em paralelo em nós distintos da rede. Seu objetivo é reduzir o tempo de execução das consultas individualmente. Por isso, o paralelismo intra-consulta é mais apropriado para obter ganho de desempenho em consultas OLAP, que são de alto custo e demandam um tempo de execução considerável (AKAL, BÖHM et al.), (RÖHM, BÖHM et al., 2000),(LIMA, 2004).

O trabalho de (RÖHM, BÖHM et al., 2000) compara duas propostas de projetos de distribuição de dados do *benchmark* TPC-R (TPC, 2005c) em um agrupamento de computadores pessoais. Em um dos projetos, propõe a replicação total, onde a base de dados inteira é replicada em todos os nós. No segundo, propõe um projeto híbrido que combina fragmentação física e replicação parcial, onde tabelas maiores são fragmentadas fisicamente e as outras tabelas são totalmente replicadas entre os nós. Os experimentos executados com consultas típicas de consultas OLAP e OLTP (*On-line Transaction Processing*) mostraram que a replicação total conseguiu aceleração linear com respeito ao número de nós e aumento da vazão de consultas executadas. Entretanto, o desempenho do projeto híbrido superou a abordagem de replicação total, alcançando aceleração superlinear. A pesquisa verificou que a fragmentação física da tabela maior em fragmentos menores fez com que o SGBD gerasse planos de execução mais

eficientes sobre essas tabelas menores, pois o tamanho reduzido da tabela permite reduzir o custo da junção além de melhor utilização do *cache* (RÖHM, BÖHM et al., 2000). Logo, a estratégia de reduzir o tamanho das tabelas maiores através da fragmentação física aliado ao paralelismo intra-consulta se mostra como uma ótima alternativa para melhorar o desempenho de consultas típicas de ambiente OLAP.

Entretanto, poucos trabalhos levam em consideração a forma como os dados são organizados em *data warehouses* durante o projeto de fragmentação. Nossa proposta é aproveitar a característica da modelagem multidimensional e utilizar um índice multidimensional para fragmentar fisicamente a tabela fato de um esquema estrela com o objetivo de reduzir o tempo de processamento de consultas OLAP.

A modelagem multidimensional é a técnica mais utilizada por projetos de *data warehouses*. O esquema de dados produzido por essa técnica é conhecido como esquema estrela. O esquema estrela é formado por dois tipos de tabela: tabelas fato e dimensão. A tabela fato armazena as medidas numéricas do negócio modelado e as chaves estrangeiras das dimensões que a descrevem. A organização dos dados de um esquema estrela pode ser vista como um espaço multidimensional conhecido como "cubo de dados" (GARCIA-MOLINA, ULLMAN et al., 2001). As arestas do cubo representam as dimensões e a combinação de dimensões define um ponto do espaço multidimensional e representa uma tupla da tabela fato.

Os índices multidimensionais por sua vez dividem um espaço multidimensional em sub-regiões para tornar mais eficiente a recuperação de objetos multidimensionais. Nossa estratégia consiste em utilizar como índice multidimensional a árvore R (árvore de região) (GUTTMAN, 1984) para particionar o espaço definido por uma tabela fato e suas dimensões relacionadas. Posteriormente, a própria árvore R é utilizada para descobrir o conjunto de fragmentos que possuem tuplas relevantes para determinada consulta. Dessa forma, nossa abordagem permite reduzir o número de fragmentos a serem processados por determinada consulta além de permitir utilizar paralelismo intra-consulta para processamento de consultas e assim atingir nosso objetivo.

A avaliação da nossa proposta foi feita através da implementação de um protótipo com a base de dados e consultas do TPC-H (TPC, 2005a) em um ambiente distribuído com 8 nós cada um executando uma instância do SGBD PostgreSQL (POSTGRESQL, 2006). Os resultados mostram cenários em que obtivemos ganho tanto em ambientes monoprocessados como em ambientes paralelos.

A dissertação está organizada da seguinte forma: no capítulo 2 apresentamos a revisão da literatura com os trabalhos relacionados. No capítulo 3 apresentamos nossa proposta de fragmentação física e o capítulo 4 apresenta os resultados de desempenho obtidos com nossa técnica. Finalmente, o capítulo 5 conclui nossa dissertação e apresenta as possibilidades de trabalhos futuros.

2. Indexação Multidimensional e Fragmentação Física

O objetivo deste capítulo é apresentar os conceitos básicos que são necessários para o entendimento da solução nos quais nossa proposta é baseada. Iniciamos o capítulo falando a respeito das características da modelagem multidimensional na seção 1. Na seção 2 descrevemos os conceitos da indexação multidimensional onde abordamos em especial a árvore R. Na seção 3 apresentamos o algoritmo de clusterização que utilizamos para agrupar os objetos na árvore R. Em seguida, na seção 4 abordamos sobre sistemas de bancos de dados distribuídos e suas vantagens. Na seção 5 discorremos sobre projeto de distribuição de dados, que inclui o projeto de fragmentação (física ou virtual) e o projeto de alocação dos fragmentos em um ambiente distribuído. Nesta seção também abordamos o processamento de consultas em paralelo propiciado pelo projeto de distribuição de dados além dos projetos de distribuição específicos para *data warehouses*. Na seção 6 analisamos alguns trabalhos relacionados e finalmente na seção 7 concluímos o capítulo.

2.1.Modelagem Multidimensional

Um *data warehouse* (DW) é um repositório de dados projetado para apoiar o processo de tomada de decisão nos negócios de uma organização (INMON, 2005). Ele acumula o histórico de dados gerado pelas transações operacionais da organização. Através de consultas feitas sobre esse repositório, os analistas de negócio podem extrair diversas informações sobre a organização, como por exemplo, o perfil de seus clientes e de sua concorrência e o desempenho dos negócios da organização nos últimos meses.

Os principais requisitos de um DW é prover facilidade de acesso a informação organizacional, garantir que os dados estejam consistentes e ser flexível a mudanças através de fácil adaptação, sendo dessa forma a base para uma constante melhora do processo de tomada de decisão da organização (KIMBALL e ROSS, 2002). O modelo dimensional se encaixa nesse contexto na medida em que organiza os dados em um formato que provê facilidade de entendimento e de absorção de evoluções provenientes do banco operacional.

Atualmente, o esquema dimensional é amplamente utilizado, sendo a técnica mais viável para apresentar, armazenar e acessar dados para um DW. O objetivo da modelagem dimensional é dispor os dados de maneira simples e de fácil compreensão.

Além disso, o esquema estrela é projetado de forma a suportar consultas que contenham operações de agregação, *roll-ups* e *drill downs* utilizados pelas aplicações OLAP.

O esquema de dados utilizado para dar suporte às transações operacionais é geralmente modelado através do modelo entidade-relacionamento (ER), composto por diagramas que representam o relacionamento entre as tabelas. A modelagem dimensional também pode ser representada através do modelo ER, pois consiste de tabelas relacionais interligadas. A diferença chave entre a modelagem utilizada no ambiente operacional e a modelagem dimensional é o nível de normalização (KIMBALL e ROSS, 2002). Na modelagem apropriada ao ambiente operacional, os dados são altamente normalizados, ou seja, eles são divididos em várias entidades com o objetivo de eliminar redundância. Essa modelagem é muito boa para o processamento operacional, pois as operações de atualizações e inserções só precisam ser feitas em um local do esquema de dados. Entretanto, ela não se encaixa nos requisitos do DW, pois o modelo altamente normalizado geralmente não é intuitivo para o usuário e não permite processar eficientemente as consultas típicas de DW.

O modelo dimensional contém a mesma informação que o modelo normalizado, mas os dados são organizados de forma que garanta desempenho de consultas, flexibilidade para evoluções e facilidade de entendimento para o usuário (KIMBALL e ROSS, 2002).

Quando o modelo dimensional é baseado em um banco de dados relacional, seu conjunto de tabelas é referido como esquema estrela. O esquema estrela é composto por dois tipos de tabelas: a tabela fato e a tabela dimensão. A tabela fato é a tabela central do modelo dimensional. Ela armazena as medidas numéricas do negócio. Cada linha da tabela fato representa uma medida numérica do negócio modelado. Cada medida numérica é descrita por um conjunto de dimensões que determina o nível de granularidade da tabela fato e contextualiza a métrica em questão. A figura 1 ilustra um exemplo de esquema estrela, que descreve as vendas diárias dos produtos de uma rede de lojas. A tabela central é a tabela fato descrita pelas dimensões produto, loja e tempo.

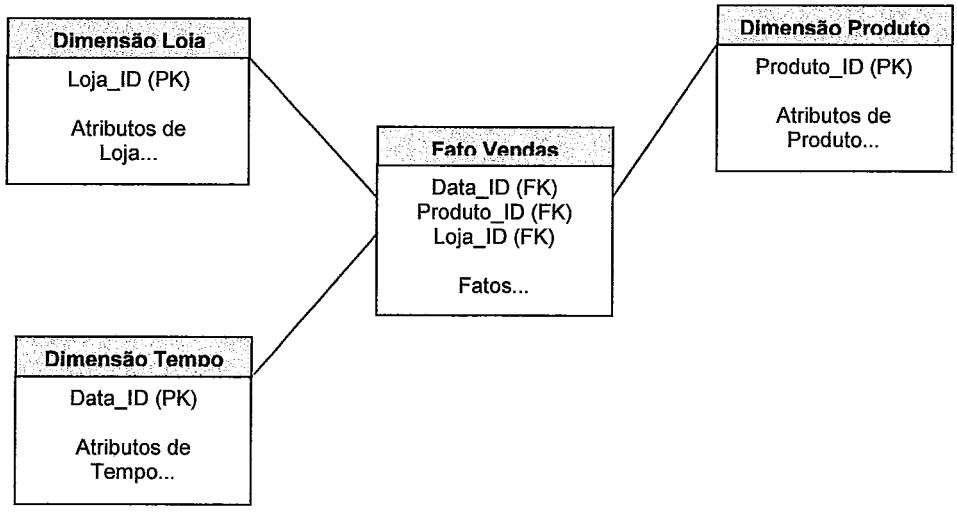

Figura 1 – Modelo dimensional (KIMBALL e ROSS, 2002)

As tabelas fato possuem uma ou mais chaves estrangeiras conectadas às chaves primárias das dimensões as quais se refere. A chave primária da tabela fato é geralmente composta pelo conjunto dessas chaves estrangeiras. As tabelas fatos expressam relacionamentos muitos para muitos.

As tabelas dimensão contêm as descrições textuais no negócio. Seus atributos são utilizados em restrições de consultas, agrupamentos e descritores de campos de relatórios. Geralmente possuem muitos atributos, mas comparadas com a tabela fato não são muito populosas. Cada dimensão possui uma chave primária que é referenciada por uma chave estrangeira de uma tabela fato.

As medidas numéricas, também conhecidas por fatos, representam os objetos de análise. Cada medida numérica depende de um conjunto de dimensões, responsáveis em dar contexto à medida. Um exemplo de medida numérica seria a quantidade vendida e suas dimensões associadas poderiam incluir o produto vendido, a loja e a data em que a venda foi feita. As tuplas da tabela fato representam a interação entre as tuplas do conjunto de tabelas dimensão, ou seja, as tuplas da tabela fato refletem os dados da transação que envolve todas as dimensões participantes.

As tabelas dimensões freqüentemente representam relacionamentos hierárquicos do negócio. Por exemplo, na tabela dimensão produto, os produtos são agrupados por marca e categoria. Cada linha armazena a marca e a categoria referente ao produto em questão. Logo, a informação hierarquizada é armazenada de forma redundante para garantir bom desempenho das consultas. Por isso as tabelas dimensões são altamente desnormalizadas.

Os atributos das tabelas dimensão podem ser vistos como dimensões de um espaço multidimensional, conhecido como "cubo de dados", com as arestas representando as dimensões produto, loja e tempo como ilustrado na figura 2. Um ponto dentro do cubo representa a medida numérica referente à combinação dos valores dessas dimensões. Logo, cada tupla da tabela fato corresponde a um ponto no espaço multidimensional (GARCIA-MOLINA, ULLMAN et al., 2001). As consultas típicas de DW agrupam os dados ao longo das dimensões desejadas através de operações de agregação. Um exemplo desse tipo de consulta é "forneça o valor total das vendas de sapato preto para cada loja e cada semestre dos anos de 2005 e 2006. No cubo, as consultas são como operações que cortam em cubos (*slice and dice)* os dados ao longo das dimensões do cubo.

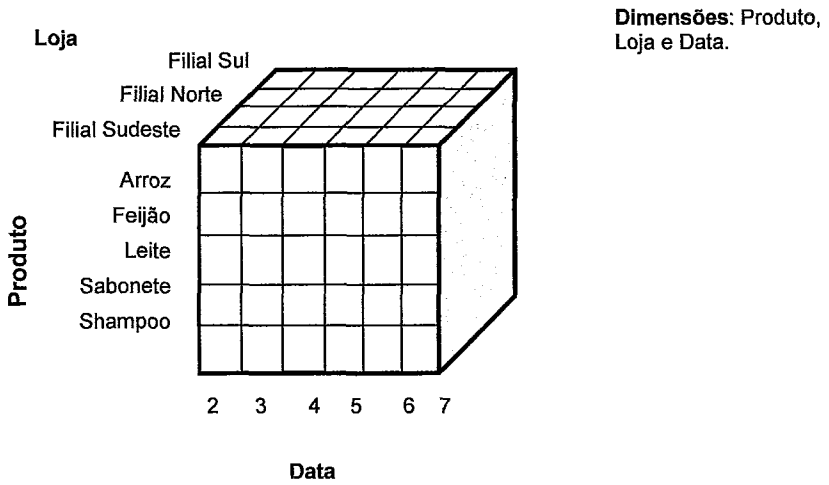

Figura 2 – Cubo de Dados

A simplicidade do modelo dimensional permite que otimizadores de banco de dados processem a consulta mais eficientemente, pois a desnormalização dos dados diminui a quantidade de junções necessárias. Além disso, sua simplicidade também permite que os usuários entendam e naveguem pelo modelo mais facilmente. Freqüentemente os analistas de negócio reconhecem seu negócio no modelo multidimensional final. Por essas razões, a modelagem dimensional é frequentemente utilizada em projetos de DW.

Na próxima seção apresentamos os conceitos básicos da indexação multidimensional e apresentamos em particular a árvore R, uma estrutura de dados muito utilizada para indexação multidimensional.

2.2. Indexação Multidimensional e a Árvore R

A principal idéia da indexação multidimensional é melhorar o acesso para recuperação de dados multidimensionais, evitando consultar todos os elementos de um espaço multidimensional para determinar aqueles que devem ser realmente recuperados. Para isso, as estruturas de indexação multidimensional organizam o espaço em sub-regiões baseando-se principalmente em duas idéias de indexação espacial: o particionamento do espaço baseado em regiões disjuntas ou aglomeração baseada em proximidade espacial, ou seja, em regiões não necessariamente disjuntas.

Geralmente o particionamento do espaço é feito de modo que as sub-regiões não possuam mais que um determinado número de pontos. Esse número máximo de pontos é geralmente a capacidade da página de disco, ou seja, o número de registros de dados que uma página é capaz de armazenar. Logo, inserções de novos pontos podem resultar em divisão de uma região em novas regiões (disjuntas ou não). Essa divisão é feita criando um ou mais hiperplanos que subdividem a região

A árvore R (árvore de retângulos) (GUTTMAN, 1984) é uma estrutura de dados inspirada na árvore B^+ para pontos ou regiões multidimensionais. Assim como a árvore B^+, ela é uma estrutura balanceada, ou seja, todos os nós folhas estão presentes no mesmo nível. Além disso, garantem utilização de pelo menos uma fração fixa do espaço, normalmente menos que 50%. Ela foi projetada pra representar pontos agrupados em regiões de duas ou mais dimensões, chamadas de hiper-retângulos. Logo, a árvore R organiza um espaço multidimensional em hiper-retângulos que agrupam objetos espacialmente próximos.

A estrutura de dados consiste de nós intermediários e nós folhas. Cada nó da árvore R possui uma quantidade variável de entradas, com tamanho máximo e mínimo pré-definidos, com exceção da raiz. Os nós intermediários são responsáveis pela indexação e cada entrada armazena um ponteiro para o nó filho e o hiper-retângulo mínimo que engloba totalmente todas as entradas do nó filho correspondente. Os nós folhas armazenam os hiper-retângulos mínimos que representam os objetos multidimensionais (GUTTMAN, 1984). A Figura 4 mostra a estrutura de uma árvore R, onde os retângulos pontilhados representam as sub-regiões. As sub-regiões não cobrem todo o espaço, o que implica que as regiões de dados que residem nas regiões maiores estão inteiramente contidas dentro de uma das regiões menores. A figura ilustra também

a sobreposição que pode ocorrer entre as regiões, embora o ideal seja minimizar as sobreposições.

O algoritmo de busca proposto por (GUTTMAN, 1984) decompõe o espaço de busca em sub-regiões e desce na árvore eliminando as regiões irrelevantes do espaço indexado até os nós folhas, onde encontra os objetos desejados. O algoritmo recebe como entrada um hiper-retângulo e o nó onde a busca deve ser feita. O objetivo é encontrar todos os registros cujo hiper-retângulo possui interseção com o hiper-retângulo de busca. A busca inicia-se pela raiz, onde cada entrada é examinada para determinar aquelas que possuem interseção com o hiper-retângulo de busca. Para todas as entradas que possuem interseção, o algoritmo de busca é aplicado novamente nos seus nós filhos e assim sucessivamente. Caso o nó examinado seja um nó folha, o algoritmo retorna a entrada como um dos resultados.

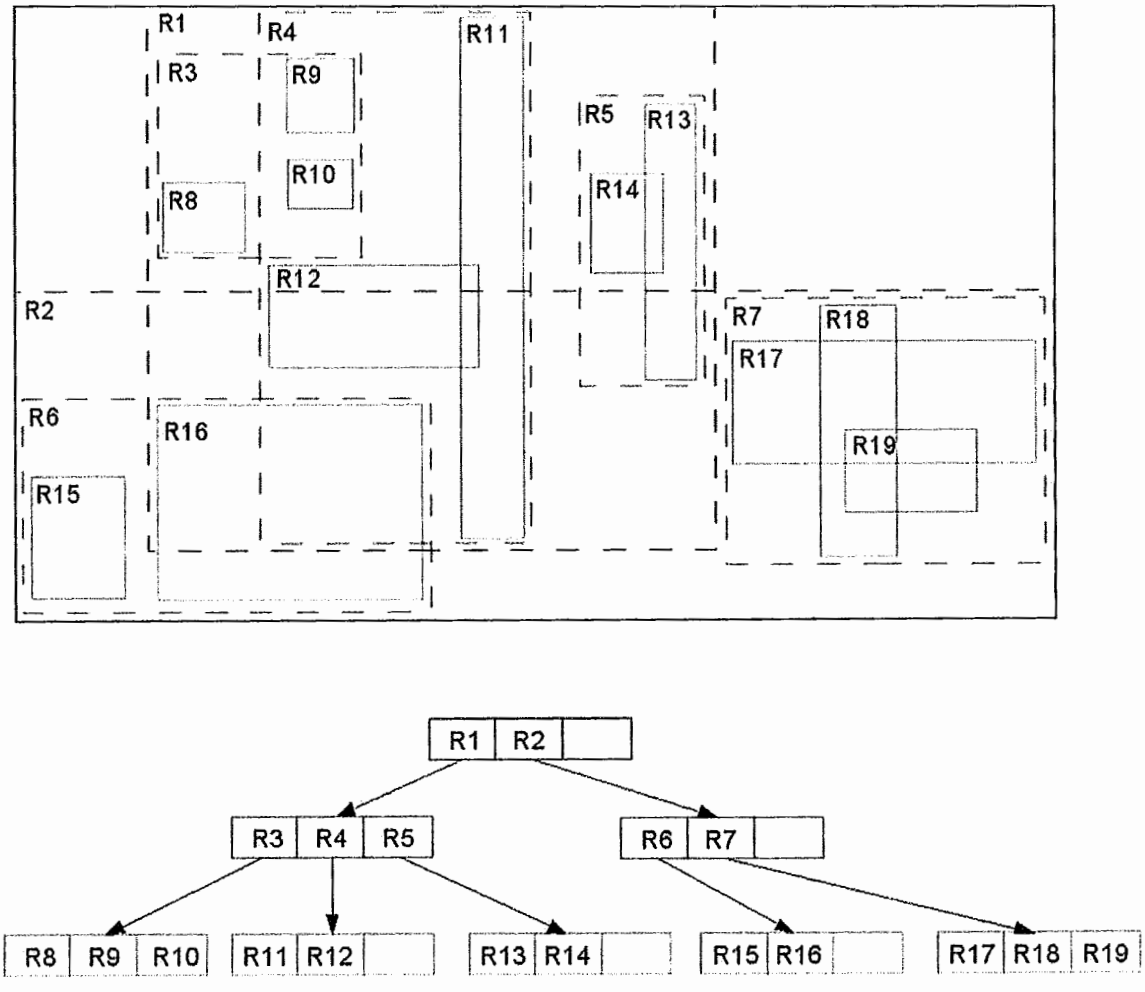

Figura 3 – Árvore R de duas dimensões.

Os algoritmos de inserção e remoção de uma entrada da árvore R utilizam os hiper-retângulos para assegurar que elementos "próximos" serão alocados no mesmo nó folha. Na inserção, as novas entradas são armazenadas nas folhas. Para escolher em que nó folha a nova entrada será armazenada, o algoritmo escolhe sempre um nó intermediário cujo hiper-retângulo necessita aumentar menos sua área para englobar a nova entrada. Esse processo é feito até que se alcance um nó folha. Se o número de entradas no nó folha ultrapassar o limite máximo de entradas definido (*overflow*), o nó sofre divisão, que consiste em dividir o nó com M + 1 entradas em dois nós, onde M é o número máximo de entradas do nó, de forma que diminua a chance de que os dois nós precisem ser examinados em buscas futuras. Para isso, o ideal é minimizar a área total dos hiper-retângulos resultantes da divisão. A divisão de um nó é propagada para os nós acima na árvore, podendo causar outras divisões até o nó raiz.

A remoção de uma entrada da árvore R ocorre através da busca do nó folha que contém a entrada para então removê-la do nó. Todo o caminho da raiz até o nó folha que sofreu a remoção deve ser atualizado. Essa atualização pode fazer com que um dos nós sofra *underflow*, ou seja, seu número total de entradas ser menor que o limite mínimo de entradas definido para a árvore. Se o *underflow* ocorrer na raiz, seu filho torna-se a nova raiz. Senão, esse nó é removido da árvore e suas entradas são reinseridas na árvore. Outra solução é redistribuir as entradas do nó entre os irmãos, de modo análogo ao que é feito na árvore B.

Em caso de estouro do número de entradas de um nó, sua divisão pode ser feita utilizando-se diferentes heurísticas para otimização do espaço. A árvore R busca minimizar a área dos nós filhos resultantes da divisão. Uma estrutura variante da árvore R é a árvore R*. Sua principal diferença é a heurística utilizada para inserção de uma nova entrada. Além de minimizar a área dos hiper-retângulos dos nós, ela também busca reduzir a margem e a sobreposição dos retângulos. (BECKMANN, KRIEGEL et al., 1990).

Nós utilizamos outra variante da árvore R proposta por (BRAKATSOULAS, PFOSER D. et al., 2002), que utiliza o algoritmo de clusterização de dados chamado k-médias para fazer a divisão de um nó que sofreu overflow. Na próxima seção falamos a respeito desse método de clusterização e suas vantagens

2.3.k-Médias

Os algoritmos de clusterização de dados são abordagens utilizadas em operações de particionamento de dados (HUANG, 1998). Os métodos de clusterização permitem particionar um conjunto de objetos em agrupamentos de forma que objetos do mesmo grupo possuam mais similaridade que objetos de outros grupos segundo algum critério pré-definido. A maior parte dos métodos de agrupamento é baseada em medidas de semelhanças ou medidas de diferença entre objetos.

Um exemplo de método para agrupar objetos é o exaustivo, que enumera todas as possíveis partições, mas sua complexidade é exponencial. Uma proposta mais viável são os métodos heurísticos, entre eles destacamos o *k-Means* (também chamado de k-Médias)

O k-Médias é um famoso algoritmo de clusterização apresentado por (MACQUEEN, 1967) muito utilizado em mineração de dados. Seu principal objetivo é classificar informações de acordo com os próprios dados, baseada em análise e comparações entre valores numéricos dos dados. Dado um conjunto de objetos numéricos e um valor inteiro k, o algoritmo busca particionar o conjunto de objetos em k grupos.

Para gerar os agrupamentos, o algoritmo calcula o centro de cada grupo, chamado de centróide e compara cada objeto com o centróide através de uma medida de distância ou similaridade, como por exemplo, a distância euclidiana ou distância de Manhattan. Os centróides são calculados pela média aritmética dos valores de cada atributo de cada objeto pertencente ao seu grupo.

O algoritmo é iniciado escolhendo-se k valores distintos para os centróides. A escolha pode ser feita aleatoriamente. Depois, todos os objetos do espaço a serem agrupados devem ser associados ao centróide mais próximo, formando agrupamentos iniciais. Para isso é preciso calcular a distância de cada ponto a todos os centróides. O próximo passo consiste em refinar o valor dos centróides, recalculando os centróides de cada um dos grupos e novamente associar os pontos aos novos centróides mais próximos. Esse último passo deve ser executado até que não haja mais alterações dos grupos, ou seja, se nenhum objeto for incorporado a um novo agrupamento diferente daquele ao qual pertencia antes. O algoritmo sempre converge a uma solução ótima local, entretanto não é garantido que alcance o ótimo global. A qualidade do resultado final do algoritmo depende dos valores escolhidos como centróides iniciais.

Uma das principais vantagens do algoritmo k-Médias é a capacidade de processar eficientemente grandes conjuntos de dados.

Nessa dissertação, utilizamos uma variante da árvore R que utiliza o k-médias para fazer a divisão do nó em caso de *overflow*. Essa variante foi proposta em (BRAKATSOULAS, PFOSER D. et al., 2002), onde eles propõe a substituição das heurísticas tradicionais utilizadas pelos algoritmos de divisão do nó pelo algoritmo de clusterização. Os resultados mostraram um aumento geral no desempenho da árvore R, com destaque para o tempo de inserção.

2.4. Sistemas de Banco de Dados Distribuído

Sistemas de computação distribuída consistem em um conjunto de elementos de processamento independentes que estão interconectados por uma rede de computadores e que cooperam entre si na execução de suas tarefas atribuídas (ÖZSU e VALDURIEZ, 1999). Os elementos de processamento são dispositivos de computação capazes de executar um programa e o conjunto desses elementos pode ser heterogêneo. O processamento distribuído surgiu da necessidade em resolver problemas grandes e complexos de maneira mais eficiente. É uma variante da estratégia "dividir para conquistar".

A idéia do processamento distribuído é dividir um problema complexo em problemas menores, mais simples de serem resolvidos. Esses fragmentos do problema complexo podem ser processados por diferentes elementos de processamento que formam um sistema onde colaboram de forma eficaz para a execução de uma tarefa comum. O fato de o problema ser dividido em partes e seu processamento ser feito por elementos distintos possibilita que ele seja resolvido paralelamente entre os componentes do sistema distribuído, o que permite um ganho maior de desempenho.

O processamento distribuído possui várias vantagens em relação ao processamento seqüencial. Os sistemas distribuídos são um método mais econômico para alcançar maior poder computacional, pois é menos custoso interconectar vários processadores do que adquirir um supercomputador, com arquitetura paralela e processadores fortemente acoplados, caros tanto em relação ao hardware quanto aos softwares específicos. Além disso, sistemas distribuídos são mais flexíveis do que máquinas isoladas, pois permitem um crescimento incremental através da inclusão de novos dispositivos computacionais para maiores desempenhos.

Os bancos de dados distribuídos são uma coleção de vários bancos de dados logicamente inter-relacionados, conectados por uma rede de computadores. Por sua vez, um sistema de banco de dados distribuído é o sistema que gerencia o banco de dados distribuído e mantém a localização dos dados transparente para as aplicações e seus usuários. Nesses sistemas, a distribuição física dos dados significa que a comunicação entre eles é feita através de uma rede, o único recurso compartilhado, e não através de memória compartilhada por exemplo (ÖZSU e VALDURIEZ, 1999).

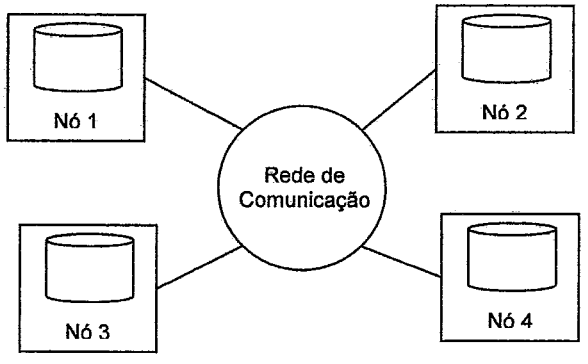

Figura 4 - Ambiente Distribuído

Um exemplo de sistema distribuído que vem ganhando destaque é o agrupamento de computadores pessoais. É considerada atualmente uma plataforma bem atraente para processamento paralelo e aceleração de transações. Eles representam uma alternativa economicamente mais interessante comparado aos servidores com arquiteturas paralelas com multiprocessadores fortemente acoplados, muito utilizados pelas aplicações que necessitam de alto poder computacional (CECCHET, MARGUERITE et al., 2004).

Um agrupamento de computadores pessoais (*cluster*) é um sistema local que consiste em um conjunto de computadores independentes interconectados em uma rede de alta velocidade (STERLING, 2001). Cada computador independente constitui um nó do agrupamento e todos os nós do agrupamento devem ser interligados por uma rede dedicada, ou seja, independente da rede que conecta o agrupamento ao mundo externo. São considerados sistemas locais, pois todos os seus componentes encontram-se fisicamente no mesmo ambiente e são gerenciados como se fosse um único sistema de computador.

Estendendo o conceito de agrupamento de computadores pessoais, o agrupamento de banco de dados consiste em uma camada intermediária de software que