Uma Arquitetura de Replicação/Fragmentação para Acesso Distribuído a Data Warehouse via Web

*Cristina Dutra de Aguiar Ciferri**
*Ricardo Rodrigues Ciferri**
Departamento de Informática
Universidade Estadual de Maringá
Av. Colombo, 5.790 – CEP 87.020-900
Maringá – PR – Brasil
cdac,rrc@din.uem.br

Fernando da Fonseca de Souza
Departamento de Informática
Universidade Federal de Pernambuco
Caixa Postal 7851 – CEP 50.732-970
Recife – PE – Brasil
fdfd@di.ufpe.br

Resumo

O tema *data warehousing* engloba arquiteturas, algoritmos e ferramentas que possibilitam que dados selecionados de provedores de informação autônomos, heterogêneos e distribuídos sejam integrados em um único repositório de dados, conhecido como *data warehouse*, o qual é voltado para suporte aos processos de gerência e tomada de decisão. Permitir que o *data warehouse* seja explorado e analisado via *Web* apresenta várias vantagens, mas também introduz diversos desafios. Dentro deste contexto, este trabalho propõe uma arquitetura de replicação/fragmentação para acesso distribuído a *data warehouse* via *Web*, considerando-se um grande número de usuários. A arquitetura proposta é genérica, podendo ser adaptada a diversos tipos de aplicação e tem como principais objetivos garantir a transparência de localização e aumentar a disponibilidade dos dados do *data warehouse*.

Palavras-Chave: Banco de Dados, Internet e World-Wide-Web, Data Warehouse.

1. Introdução

O tema *data warehousing* engloba arquiteturas, algoritmos e ferramentas que possibilitam que dados selecionados de diferentes provedores de informação autônomos e distribuídos sejam integrados em um único repositório de dados, conhecido como *data warehouse*. Tais provedores de informação podem possuir uma variedade de formatos e modelos e podem incluir desde sistemas gerenciadores de banco de dados (SGBDs) relacionais, orientados a objetos e objeto-relacionais até bases de conhecimento, sistemas legados, documentos HTML e SGML, dentre outros [Wid95, Wie97].

O acesso à informação integrada dos diferentes provedores de informação é realizada, geralmente, em duas etapas: (1) a informação de cada provedor pode ser extraída previamente, devendo ser traduzida, filtrada, integrada à informação relevante de outros provedores e finalmente armazenada no *data warehouse*; e (2) as consultas, quando realizadas, são executadas diretamente no repositório, sem acessar os provedores de informação originais. Desta forma, a informação integrada torna-se disponível para consulta ou análise imediata de usuários finais e programas de aplicação.

Permitir que o *data warehouse* seja explorado e analisado via *Web* apresenta várias vantagens, tais como facilitar o acesso aos dados de provedores de informação que já apresentem interface para a *Web*, possibilitar o acesso quase universal dos clientes aos dados contidos no *data warehouse*, integrar o ambiente informacional de modernas empresas globalizadas e reduzir os custos relativos à implementação e operação. Como exemplo, [CR97] cita que uma empresa não precisará investir em uma nova estrutura do lado cliente cada vez que desejar dar acesso ao *data warehouse* a um novo usuário.

* suportado pelo programa PICDT-CAPES/UEM.

Por outro lado, a *Web* introduz novos desafios, exigindo, por exemplo, que sejam oferecidos mecanismos de comunicação eficientes e seguros que permitam a troca de informação entre os provedores de informação heterogêneos e distribuídos e o *data warehouse*, e entre o *data warehouse* e os clientes, além de modelos de interface que facilitem que usuários típicos de sistemas de tomada de decisão realizem consultas no *data warehouse* de maneira amigável. Outros desafios referem-se ao suporte a um grande número de usuários e ao oferecimento de arquiteturas que permitam que o sistema permaneça operacional frente a falhas parciais [SSU96].

Dentro deste contexto, este artigo apresenta uma arquitetura de replicação/fragmentação para acesso distribuído a *data warehouse* via *Web*. Visa, desta forma, dar suporte à troca de informações entre o *data warehouse* e os clientes, considerando-se um grande número de usuários. Dentre os objetivos desta arquitetura, pode-se citar: aumentar a disponibilidade dos dados do *data warehouse*, aumentar a disponibilidade de acesso ao *data warehouse*, garantir transparência de localização, prover aumento de desempenho no processamento de consultas e garantir portabilidade.

A arquitetura proposta é genérica, de modo a permitir sua adaptação a diversos tipos de aplicação, sendo baseada nas seguintes características:

- os dados do *data warehouse* são armazenados segundo o **esquema estrela**;
- os dados do *data warehouse* apresentam diferentes **níveis de agregação**;
- a solução apresentada utiliza os conceitos de fragmentação e replicação presentes em sistemas de banco de dados distribuídos e características especiais de *data warehousing*;
- a solução apresentada não é baseada em força bruta (exclusivamente *hardware*); e
- a arquitetura segue a abordagem cliente-servidor.

O restante deste trabalho é estruturado da seguinte maneira. A próxima seção define conceitos básicos do tema *data warehousing* necessários ao entendimento da arquitetura. A seção 3 realiza uma breve descrição de trabalhos correlatos. A seção 4 apresenta a arquitetura proposta sob três enfoques: distribuição dos dados do *data warehouse*, forma na qual consultas submetidas ao *data warehouse* são redirecionadas através do **login site** e solução para o acesso aos dados. A seção 5 descreve o **WebDW**, um estudo de caso baseado na arquitetura proposta para uma cadeia de supermercados. O artigo é finalizado na seção 6 com as conclusões e extensões.

2. Data Warehouse

Em um *data warehouse*, os dados são organizados segundo diferentes **níveis de agregação** [Inm96]. O nível inferior contém dados primitivos coletados diretamente do ambiente operacional. Por outro lado, o **nível superior** da **hierarquia de agregação** possui dados altamente resumidos. Entre o nível inferior e superior podem existir vários níveis intermediários, representando graus de agregação crescente. Os dados armazenados em um nível n correspondem a alguma forma de agregação dos dados armazenados em um nível n-1, sendo que o nível intermediário mais inferior utiliza os dados do nível inferior como base para as suas agregações.

Desta forma, um usuário típico de sistemas de tomada de decisão pode iniciar sua análise no **nível superior** para obter uma visão geral do negócio, podendo percorrer a **hierarquia de agregação** até o nível inferior à medida que dados específicos sejam necessários (processo *drill-down* – processo inverso *rollup*). Por exemplo, o seguinte processo *drill-down* poderia ser solicitado: (1) vendas anuais dos produtos da marca M em todas as filiais; (2) vendas mensais no ano de 1998 dos produtos da marca M fora de promoções nas filiais 1 e 2; e (3) vendas diárias no mês de outubro de 1998 do produto P da marca M fora de promoções na filial 1.

As análises realizadas pelos usuários representam, de maneira geral, requisições multidimensionais aos dados do *data warehouse* [CCS93, Kim96, Sho97, Rou97], as quais têm por objetivo a visualização dos dados segundo diferentes perspectivas. No modelo de dados conceitual multidimensional, existe um conjunto de medidas numéricas, que são os objetos de análise relevantes ao negócio, e um conjunto de dimensões, ou variáveis, que

determinam o contexto para a medida [CD97, WB97, CT98, GMR98]. Uma medida numérica pode ser definida como uma função de suas dimensões correspondentes, representando, desta forma, um valor no espaço multidimensional. Como exemplo, a medida numérica unidades-vendidas em um *data warehouse* de uma cadeia de supermercados pode ser determinada pelas dimensões produto, promoção, filial e tempo.

Ademais, cada uma das dimensões pode ser descrita por um conjunto de atributos, sendo que os atributos de uma dimensão podem se relacionar através de **hierarquias de relacionamento**. Suponha os seguintes atributos para a dimensão tempo: mês, quarto e semestre. Exemplos de **hierarquias de relacionamento de atributos** são: mês (janeiro ... março) = 1° quarto; mês (abril ... junho) = 2° quarto; mês (julho ... setembro) = 3° quarto; mês (outubro ... dezembro) = 4° quarto; e quarto (1°, 2°) = 1° semestre; quarto (3°, 4°) = 2° semestre.

Existem duas alternativas utilizadas para a representação lógica do modelo de dados conceitual multidimensional: estruturas relacionais e bancos de dados multidimensionais (BDMs). Na primeira alternativa, dois tipos de esquema são utilizados: estrela e floco de neve.

O **esquema estrela** possui uma tabela de fatos dominante no centro do esquema e um conjunto de tabelas de dimensão nas extremidades. A tabela de fatos armazena as medidas numéricas relevantes ao negócio, além dos valores das dimensões descritivas para cada instância, os quais são responsáveis pela ligação dos fatos às diversas dimensões. Já as tabelas de dimensão armazenam dados alfanuméricos correspondentes às dimensões do negócio (atributos) e possuem uma chave para cada uma de suas instâncias.

Enquanto o **esquema estrela** apresenta uma estrutura desnormalizada, visando obter um bom desempenho para consultas altamente complexas, o esquema floco de neve apresenta uma estrutura normalizada, na qual cada extremidade do **esquema estrela** passa a ser o centro de outras estrelas, suportando assim a representação explícita da **hierarquia de relacionamento de atributos**. O uso do esquema floco de neve, entretanto, deve ser ponderado entre os ganhos em termos de espaço de armazenamento e os custos de complexidade para o usuário.

BDMs surgiram como uma segunda alternativa para a representação e o armazenamento de dados multidimensionais. Tal alternativa armazena diretamente os dados em estruturas de dados especiais (geralmente matrizes) e implementa as operações multidimensionais sobre estas estruturas. Por exemplo, as medidas numéricas podem ser representadas por matrizes multidimensionais cujos índices, os quais representam as dimensões, variam sobre segmentos contínuos dos números naturais.

A utilização de BDMs apresenta como principais vantagens a facilidade de manipulação e visualização dos dados, eficientes mecanismos de armazenamento e recuperação, tratamento de esparcidade e presença das perspectivas embutidas diretamente na estrutura. Entretanto, BDMs dificultam o processo de reestruturação de dimensões e apresentam problemas quando o número de dimensões é elevado, sendo assim indicados somente para ambientes estáveis. Já a tecnologia relacional está consolidada, permite maior capacidade de armazenamento, possui flexibilidade na reestruturação e no acesso aos dados, incorpora novas tecnologias, tal como paralelismo, e permite maior integração com dados não-numéricos. Ademais, já existem trabalhos sendo realizados no sentido de se estender a linguagem SQL para o suporte eficiente às operações multidimensionais [GBPL95, GL97, RS97].

3. Trabalhos Correlatos

Existem dois componentes principais em um ambiente de *data warehousing*: o **componente de integração**, responsável por extrair as informações relevantes dos provedores de informação que participam do ambiente operacional, traduzi-las, filtrá-las, integrá-las e armazená-las no *data warehouse* e o **componente de análise e consulta**, responsável por atender às requisições multidimensionais dos usuários finais.

Diversos trabalhos e/ou sistemas têm sido desenvolvidos enfocando cada um destes componentes (WHIPS [HGMW+95, WGL+96], Redbrick [Red99]) ou ambos (Oracle [Ora99a], IBM [IBM99]). Em adição, vários trabalhos já estão incorporando a tecnologia *Web*, tanto na integração de provedores de informação heterogêneos [FDLS97, FGLM+98, MZ98, SC98], quanto no suporte a OLAP (*On-Line Analytical Processing*) [Ora99b, IBM99].

O sistema WHIPS propõe uma arquitetura que identifica alterações nos dados de provedores de informação autônomos e heterogêneos, transforma e agrega estes dados de acordo com as especificações do sistema e os integra incrementalmente ao *data warehouse*. Enfoca, portanto, o componente de integração. Por outro lado, o sistema Redbrick enfoca o componente de análise e consulta, uma vez que armazena os dados em estruturas relacionais projetadas especialmente para acesso eficiente e oferece facilidades de consulta para usuários finais. Os produtos de *data warehousing* das empresas Oracle (Oracle Warehouse + Oracle Express OLAP) e da IBM (IBM Visual Warehouse + DB2 OLAP Server) oferecem suporte a ambos componentes de integração e de análise e consulta. Tais produtos incluem um vasto conjunto de ferramentas e de pacotes aplicativos para *data warehouse* e permitem o acoplamento de ferramentas de outras empresas voltadas para a análise de negócios. Já os trabalhos apresentados em [FDLS97, FGLM+98, MZ98, SC98] não tratam especificamente do tema *data warehousing*, mas oferecem modelos de dados e/ou linguagens de consultas para visões *Web* integradas, criadas a partir de provedores de informação heterogêneos.

Os trabalhos que enfocam o componente de integração diferem da arquitetura proposta neste artigo por tratarem da construção do *data warehouse*, ao passo que a arquitetura considera que o *data warehouse* já está construído. Assim, qualquer um destes trabalhos, além dos descritos em [FDLS97, FGLM+98, MZ98, SC98] poderia ser adaptado a arquitetura de forma a originar o *data warehouse* base a ser utilizado. Em adição, a arquitetura tem por objetivos propor uma forma de acesso e distribuição aos dados do *data warehouse* e um mecanismo para aumentar a disponibilidade de acesso aos seus dados. Desta forma, trabalhos que enfocam o componente de análise e consulta poderiam ser incorporados à arquitetura com o objetivo de estender as funcionalidades do **login site** (seção 4.2) com relação à visualização dos resultados de uma consulta. Por fim, a arquitetura apresentada neste artigo propõe a distribuição do *data warehouse* em diversos *sites*, sendo que, como resultado, cada *site* armazena um *data warehouse* local. Dentro deste contexto, os produtos Oracle e IBM poderiam dar suporte aos diversos *data warehouse* locais.

4. Arquitetura Proposta

A descrição da arquitetura será realizada em três partes. Inicialmente, será discutida a forma de distribuição dos dados do *data warehouse*. Em seguida, será descrita a forma na qual consultas submetidas ao *data warehouse* são redirecionadas através do **login site**. Por fim, será apresentada a solução para o acesso aos dados.

4.1 Distribuição dos Dados

A proposta de distribuição dos dados do *data warehouse* é baseada nos conceitos de fragmentação e replicação, organização dos dados em diferentes **hierarquias de agregação** e presença de um **data warehouse global**. Tal proposta constitui a base de um *data warehouse* distribuído, o qual faz uso da tecnologia *Web* para proporcionar um processamento eficiente de consultas e o aumento da disponibilidade dos dados.

Organização dos Dados

A arquitetura propõe que os dados do *data warehouse* sejam armazenados segundo o **esquema estrela**, com base na discussão realizada na seção 2 e no fato de que este esquema é a forma de representação e armazenamento de dados multidimensionais mais utilizada atualmente. Ademais, a arquitetura organiza os dados do *data warehouse* em um **nível inferior** e em um conjunto finito H de **hierarquias de agregação**.

O **nível inferior** representa o **esquema estrela** base do *data warehouse*, formado por dados primitivos coletados diretamente do ambiente operacional, sendo altamente volumoso e não agregado. Este nível contém uma tabela de fatos F e um conjunto D de dimensões, sendo utilizado como base por qualquer **hierarquia de agregação**.

Já cada **hierarquia de agregação** $h_i \in H$, onde $1 \leq i \leq m$, representa uma agregação do **nível inferior** com relação a uma ou mais de suas dimensões e possui k **níveis de agregação** ($\{n_1, \ldots, n_k\}$), sendo que:

- caso exista $h_i, h_j \in H$, onde $1 \leq i, j \leq m$, deve-se ter $h_i \neq h_j$, ou seja, **hierarquias de agregação** devem ser distintas entre si;
- cada $n_x \in \{n_1, \ldots, n_k\}$, onde $1 \leq x \leq k$, deve:
  - possuir as mesmas dimensões de h_i;
  - satisfazer ao critério de crescimento, o qual define que pelo menos uma dimensão do nível anterior deve ser agregada com relação a sua **hierarquia de relacionamento de atributos** (aumento da granularidade) e as demais dimensões devem possuir no mínimo o mesmo **nível de agregação** do nível anterior;
  - possuir uma nova tabela de fatos F_x gerada a partir de F_{x-1} de acordo com as agregações de h_i e um novo conjunto D_x de tabelas de dimensão que: (1) contém as tabelas de dimensão geradas a partir das agregações existentes e (2) compartilha com o nível anterior as demais tabelas de dimensão que não sofrem agregação. Para o primeiro nível de h_i ($x = 1$) tem-se $F_{x-1} = F$.

A escolha das dimensões de h_i, com base no **nível inferior**, determina qual tipo de relacionamento multidimensional quer-se observar, tais como vendas por mês, vendas por mês por loja, ou vendas por mês por loja por produto. Como exemplo, suponha que o **esquema estrela** do **nível inferior** seja formado pela tabela de fatos vendas e pelas dimensões produto (atributos: item, marca, distribuidora), filial (atributos: loja, cidade, região), e tempo (atributos: dia, mês, ano). Considerando as hierarquias de relacionamento existentes entre os atributos destas dimensões, as seguintes **hierarquias de agregação** e seus respectivos **níveis de agregação** poderiam ser gerados:

- h_1 = agregação baseada nas dimensões produto e tempo (relacionamento multidimensional vendas por produto por tempo)
  - n_1 : granularidade (produto, tempo) = (item, mês)
    $\Rightarrow$ manutenção da granularidade de produto e aumento da granularidade de tempo
  - n_2 : granularidade (produto, tempo) = (marca, mês)
    $\Rightarrow$ aumento da granularidade de produto e manutenção da granularidade de tempo
  - n_3 : granularidade (produto, tempo) = (distribuidora, ano)
    $\Rightarrow$ aumento das granularidades de produto e de tempo
- h_2 = agregação baseada nas dimensões produto e filial (relacionamento multidimensional vendas por produto por filial)
- h_3 = agregação baseada nas dimensões produto, filial e tempo (relacionamento multidimensional vendas por produto por filial por tempo)

Embora a arquitetura seja genérica, permitindo a agregação de quaisquer dimensões do **nível inferior**, [CD97] cita que a agregação pela dimensão tempo é de suma importância para sistemas de suporte à decisão, como exemplo no processo de análise de tendências.

Data Warehouse Global

A arquitetura faz uso de um **data warehouse global**, armazenado em um *site* separado, que possui todos os dados relevantes à organização, de acordo com o **esquema estrela**, o **nível inferior** e todas as **hierarquias de agregação** gerados a partir deste. Neste *site* não existe, em princípio, fragmentação ou replicação dos dados. A existência deste **data warehouse global** está relacionada ao fato de que este *site* seria capaz de responder a

qualquer consulta do nível gerencial. Este **data warehouse global** deve ser fragmentado e replicado em vários *sites*, conforme discutido a seguir.

Fragmentação

A fragmentação do **data warehouse global** implica na fragmentação das tabelas de fatos tanto do **nível inferior** quanto dos **níveis de agregação** das hierarquias. Para uma dada **hierarquia de agregação**, deve-se escolher as dimensões a serem utilizadas na fragmentação das tabelas de fatos de seus **níveis de agregação**, sendo esta fragmentação estendida à tabela de fatos do **nível inferior**.

A escolha destas dimensões deve ser realizada pelo projetista do *data warehouse*, baseada em uma análise criteriosa das necessidades do negócio e na localização física do *site* que irá armazenar os dados. Para tanto, o projetista deve considerar que existem dimensões comuns a diversas aplicações de negócios, tais como tempo, filial, cliente, departamento e produto. Dentre estas dimensões, deve-se escolher pelo menos uma que permita que porções específicas do negócio sejam analisadas separadamente. Por exemplo, em uma cadeia de supermercados, uma dimensão a ser utilizada é a dimensão filial, uma vez que cada supermercado pode armazenar a sua porção específica de dados. Ademais, usuários de uma filial tendem, na maioria das vezes, a acessar exclusivamente os dados daquela filial.

Assim, o seguinte algoritmo de fragmentação deve ser usado:

para cada $h_i \in H$, onde $1 \leq i \leq m$,

determine uma ou mais dimensões ($\{d_1, \ldots, d_j\}$) a serem utilizadas na sua fragmentação

fragmente o **esquema estrela** do **nível inferior** de acordo com $d_1, \ldots, d_j$

$\Rightarrow$ fragmente a tabela de fatos

$\Rightarrow$ fragmente as tabelas de dimensão correspondentes a $d_1, \ldots, d_j$

$\Rightarrow$ copie as demais tabelas de dimensão

para cada $n_x \in \{n_1, \ldots, n_k\}$ não replicado, onde $1 \leq x \leq k$ de h_i

fragmente seu **esquema estrela** de acordo com $d_1, \ldots, d_j$

$\Rightarrow$ fragmente a tabela de fatos

$\Rightarrow$ fragmente as tabelas de dimensão correspondentes a $d_1, \ldots, d_j$

$\Rightarrow$ copie as demais tabelas de dimensão

Vale destacar que, uma vez escolhidas as dimensões a serem fragmentadas em uma hierarquia, todos os seus **níveis de agregação** devem ser fragmentados de acordo com estas dimensões. No entanto, **hierarquias de agregação** distintas podem ser fragmentadas baseadas em dimensões distintas. Em adição, dimensões comuns podem ser compartilhadas pelas diversas tabelas de fatos geradas.

Desta forma, cada *site* possuirá a tabela de fatos do **nível inferior** e dos demais níveis das **hierarquias de agregação** fragmentadas de acordo com os atributos chave relativos às dimensões utilizadas na fragmentação de cada uma das hierarquias. O mesmo ocorre com as tabelas de dimensão correspondentes a estes atributos chave. Em adição, cada *site* possuirá uma cópia das demais tabelas de dimensão. Na arquitetura proposta, as tabelas de fatos devem ser fragmentadas horizontalmente, de forma completa [EN94], independentemente do número de dimensões utilizadas na fragmentação.

Replicação

A determinação de quais dados do **data warehouse global** devem ser replicados nos diversos *sites* é baseada na quantidade de dados armazenados em cada **nível de agregação** de cada hierarquia e na quantidade de usuários que acessam cada um destes níveis. De maneira geral, o **nível inferior** é altamente volumoso e apresenta uma menor quantidade de usuários que o acessam. Por outro lado, o **nível superior** de cada

hierarquia de agregação é pouco volumoso, além de ser acessado por uma grande quantidade de usuários. Assim, a arquitetura propõe a replicação dos **níveis de agregação** mais superiores de cada uma das hierarquias (maior grau de agregação) em todos os *sites*, ao passo que os demais níveis estão presentes somente no **data warehouse global** e seus fragmentos espalhados pelos diversos *sites*, conforme descrito anteriormente. Esta replicação minimiza o tráfego de dados na rede, uma vez que as consultas mais freqüentes são respondidas por dados destes níveis, os quais estão presentes em qualquer um dos *sites*.

Entretanto, caso necessário, o projetista do *data warehouse* pode replicar outros **níveis de agregação** de cada hierarquia (sempre os próximos níveis mais superiores). Para tanto, o projetista deve levar em consideração, além dos critérios acima destacados, as necessidades do negócio, as questões de desempenho e disponibilidade do sistema, os tipos e as freqüências das transações submetidas em cada *site*, as capacidades de processamento e armazenamento dos *sites* e a taxa de transmissão dos canais de comunicação.

Considerações

A arquitetura proposta considera que o *data warehouse* já está construído, e portanto não será atacado o problema de carregamento adicional de informações a partir de provedores de informação, o qual envolve as fases de extração, tradução, filtragem e integração de informações. Assim, problemas de otimização com relação ao gerenciamento de visões materializadas, tais como quais visões devem ser materializadas [BPT97, YKL97, SDN98] e manutenção das visões [MQM97, Huy97] estão fora do escopo deste artigo.

4.2 *Login Site*

Com o propósito de facilitar a comunicação entre os usuários e o *data warehouse*, a arquitetura proposta faz uso de um tipo especial de *site*, chamado **login site**, que oferece uma interface via *Web* através da qual usuários típicos de sistemas de tomada de decisão podem realizar consultas ao *data warehouse* de forma transparente. Dentre as funcionalidades do **login site**, pode-se citar: (a) autenticação de usuários, (b) oferecimento de uma interface que realiza a tradução automática de consultas gráficas em comandos de acesso ao *data warehouse*, (c) redirecionamento de consultas para os *sites* adequados e (d) gerenciamento de consultas distribuídas.

O **login site** é gerenciado por um módulo especial, o **gerenciador de login sites**, o qual é responsável por controlar a replicação do **login site**. A possibilidade de tal replicação tem por objetivo evitar problemas de gargalo no acesso a um único **login site** centralizado, contribuindo, desta forma, para o aumento da disponibilidade de acesso aos dados do *data warehouse*.

Adicionalmente à funcionalidade de replicação, o **gerenciador de login sites** também é responsável em gerenciar **wrappers** em cada um dos *sites* que contêm dados do *data warehouse*. Um **wrapper** controla o acesso local aos dados de um *site* e difere do **login site** por apresentar funcionalidade direcionada. Assim, somente as funcionalidades (a) e (b) do **login site**, com escopo local, são suportadas. O gerenciamento de **wrappers** locais permite o aproveitamento da proximidade dado/usuário, fazendo com que consultas locais efetuadas por usuários locais sejam respondidas no próprio *site*, diminuindo assim a sobrecarga na rede e no(s) **login(s) site(s).**

Interface e Tradução Automática

Para facilitar a obtenção de informações estratégicas, o **login site** deve disponibilizar uma interface gráfica, via *Web*, que permita que usuários finais selecionem o tipo de relatório (tipo de relacionamento multidimensional), a granularidade do relatório (granularidade das dimensões envolvidas, ou seja, qual o **nível de agregação** de uma dada hierarquia) e a porção específica do negócio envolvida na consulta (baseado no critério de fragmentação). Esta interface facilita, portanto, o suporte às requisições multidimensionais.

Visando o acesso às informações do *data warehouse* por usuários que em sua grande maioria não possuem tempo e/ou conhecimento para utilizar uma linguagem de consulta de banco de dados, a arquitetura propõe a

tradução automática da consulta do usuário para o padrão SQL e a posterior submissão desta para os *sites* apropriados em respondê-la. Vale destacar que uma consulta típica de *data warehousing* envolve apenas dados provenientes de um mesmo nível de uma **hierarquia de agregação**. Caso o usuário esteja percorrendo os diversos níveis da hierarquia de acordo com o processo *drill-down*, cada acesso a um nível distinto deve originar uma nova consulta. Como resultado a uma consulta são gerados gráficos e/ou tabelas que facilitam a comparação e a identificação de tendências e padrões, de acordo com sistemas OLAP.

Redirecionamento de Consultas

Esta funcionalidade apresentada pelo **login site** tem por objetivo determinar quais *sites* são adequados para responder à consulta solicitada. A determinação destes *sites* é baseada nos seguintes critérios de seletividade: disponibilidade de dados para atender à consulta, disponibilidade do *site* (*alive*), número de conexões ativas no *site*, taxa de transmissão do canal de comunicação e capacidade de processamento do *site*. Tais critérios devem ser ponderados baseados nas necessidades da aplicação envolvida, sendo os critérios de disponibilidade excludentes. Esta funcionalidade é realizada de maneira a garantir a transparência de localização dos dados.

Na arquitetura proposta, uma consulta pode ser processada tanto de maneira centralizada quanto distribuída. Tal escolha deve proporcionar um processamento eficiente da consulta e para isto devem ser considerados os fatores relativos ao **nível de agregação** dos dados e à quantidade de fragmentos, além dos critérios de seletividade acima apresentados.

Os itens a seguir apresentam as regras para o processamento/redirecionamento das consultas:

- consultas envolvendo dados do **nível superior** de qualquer **hierarquia de agregação**, independentemente da quantidade de fragmentos, podem ser efetuadas em qualquer *site* de forma centralizada;
- consultas envolvendo dados do **nível inferior** e apenas um único fragmento podem ser realizadas no *site* que possui o fragmento ou no *site* do **data warehouse global**, de forma centralizada;
- consultas envolvendo dados do **nível inferior** e vários fragmentos devem ser realizadas de maneira centralizada no *site* do **data warehouse global**;
- consultas envolvendo dados dos demais níveis e apenas um único fragmento podem ser realizadas no *site* que possui o fragmento ou no *site* do **data warehouse global**, de forma centralizada; e
- consultas envolvendo dados dos demais níveis e vários fragmentos podem ser realizadas de maneira centralizada no *site* do **data warehouse global** ou de forma distribuída nos *sites* que possuem os fragmentos.

De maneira resumida, não existe consulta distribuída em três diferentes situações: quando se considera dados provenientes de apenas um único fragmento, quando um ou mais fragmentos do **nível superior** de qualquer **hierarquia de agregação** são acessados e quando o **nível inferior** é acessado. A explicação para as duas primeiras situações é trivial, uma vez que os dados necessários para responder à consulta estão disponíveis em um mesmo *site*. Já toda consulta que envolve vários fragmentos do **nível inferior** é realizada de maneira centralizada visando a otimização da mesma. Tais consultas manipulam uma quantidade extremamente grande de dados para processar suas respostas e o tráfego de resultados parciais pela rede seria inviável. Neste caso, a disponibilidade dos dados do **data warehouse global** poderia ser melhorada duplicando-se o *site*.

A única situação na qual uma consulta pode ser processada tanto de maneira centralizada quanto distribuída é quando esta envolve dados dos demais níveis (não relativos ao **nível inferior** ou a um **nível superior** de qualquer **hierarquia de agregação**) e acessa vários fragmentos. Neste caso, a escolha da maneira na qual a consulta deve ser processada depende, além dos critérios de seletividade anteriormente apresentados, da quantidade de dados referentes aos resultados parciais que trafegarão pela rede. Este custo deve ser ponderado com o objetivo de maximizar o desempenho.

Gerenciamento de Consultas Distribuídas

No gerenciamento de consultas distribuídas, o **login site** é responsável por receber uma consulta e decompô-la em subconsultas a serem enviadas e processadas parcialmente em cada um dos *sites* envolvidos, até a obtenção da resposta à consulta inicial. Para tanto, utiliza a técnica de *semijoin* [EN94]. Vale destacar que o **login site** é responsável pelo gerenciamento da consulta distribuída e pelo envio da resposta ao usuário que a solicitou, ao passo que o processamento da mesma é realizado nos *sites* nos quais as subconsultas são enviadas. Por outro lado, quando a consulta for realizada de maneira centralizada, então o *site* que a processou se encarregará de enviar a resposta diretamente para o usuário, diminuindo a sobrecarga do **login site**.

4.3 Acesso aos Dados

O acesso aos dados do *data warehouse* segue a arquitetura *three-tier* [Ree97], segundo o paradigma de orientação a objetos. Nesta arquitetura estão presentes três componentes: clientes, servidor de aplicação (ou de objetos) e servidor de dados. Os clientes possuem a camada de apresentação, responsável pelo código da interface com os usuários finais. Tal camada não se preocupa com a forma e localização dos dados, interagindo somente com o servidor de aplicação através da requisição de objetos. Desta forma, mudanças na estrutura do banco de dados não requerem alterações na interface da aplicação.

O servidor de aplicação representa uma camada intermediária que isola o processamento de dados e disponibiliza um conjunto de objetos que encapsulam as regras de negócio, as quais são relacionadas ao problema sendo tratado. A interface de armazenamento dos dados é escondida de objetos externos, sendo de responsabilidade do servidor de aplicação o tratamento transparente do relacionamento existente entre objetos e dados e a comunicação com o servidor de dados. O servidor de dados, por sua vez, é responsável pelo armazenamento e recuperação de dados segundo as funcionalidades presentes em SGBDs, tais como gerenciamento de transações, processamento e otimização de consultas.

Na arquitetura proposta, os servidores de aplicação e de dados estão presentes em cada um dos *sites*, enquanto que os clientes possuem apenas a funcionalidade de visualização dos resultados, a qual é garantida pela camada de apresentação. O uso da arquitetura *three-tier* permite o crescimento do número de usuários e da quantidade de dados armazenados no *data warehouse*.

5. Estudo de Caso

O sistema **WebDW** (***Web-based Data Warehousing***) consiste em um estudo de caso baseado na arquitetura proposta, e foi implementado no Departamento de Informática da Universidade Estadual de Pernambuco, em ambiente UNIX, em estações de trabalho SUN. A implementação foi realizada utilizando-se a linguagem de programação Java versão 1.1.5 e o API JDBC versão 1.1 para comunicação com o SGBD SQL Server, sendo este responsável pelo armazenamento e recuperação dos dados dos *sites*. O sistema encontra-se disponível em http://www.di.ufpe.br/~cdac/webdw.

A aplicação utilizada no estudo de caso consiste em uma cadeia de supermercados e foi adaptada a partir de um exemplo real proposto por [Kim96]. Cada supermercado da cadeia é completo, e possui departamentos responsáveis pela mercearia, comida congelada, produtos derivados do leite, açougue, padaria, produtos de limpeza, bebidas, floricultura, vestuário, e farmácia, dentre outros. A aplicação armazena detalhes de quais produtos são vendidos em quais supermercados, com quais preços, em quais dias e sob quais promoções. Foram considerados 60 produtos, 11 promoções, 2 supermercados e o período de tempo de outubro de 1994 a dezembro de 1994 e outubro de 1995 a dezembro de 1995.

O **esquema estrela** desta aplicação é formado pela tabela de fatos vendas e pelas dimensões produto, filial, promoção e tempo, sendo que cada instância da dimensão filial corresponde a um supermercado da cadeia. Os principais atributos destas tabelas são:

- **vendas**: ch_produto, ch_filial, ch_promoção, ch_tempo, vendas_dolar, unidades_vendidas, custo_dolar;
- **produto**: ch_produto, descrição, descrição_completa, marca, subcategoria, categoria, departamento, tipo_pacote, tamanho_pacote;
- **filial**: ch_filial, nome_filial, número_filial, endereço_filial, cidade, estado, código_postal, distrito_vendas, região_vendas;
- **promoção**: ch_promoção, nome_promoção, tipo_redução_preço, tipo_anúncio, custo, data_início, data_fim; e
- **tempo**: ch_tempo, dia_semana, número_dia_mês, número_dia_geral, número_semana_ano, número_semana_geral, mês, quarto, período_fiscal, indicador_feriado, data, ano.

A seguir é feita uma breve descrição do estudo de caso, com relação aos demais itens da seção 4.1:

- **nível inferior**: representa o **esquema estrela** acima com granularidade diária. Permite, desta forma, a análise de produto por supermercado por promoção por dia
- **hierarquia de agregação h_1**: baseada na agregação das dimensões produto, filial, promoção e tempo
  - n_1: granularidade (produto, filial, promoção e tempo) = (produto, filial, promoção, mês) $\Rightarrow$ produto por filial por promoção por mês
  - n_2: granularidade (produto, filial, promoção e tempo) = (produto, filial, promoção, ano) $\Rightarrow$ produto por filial por promoção por ano
- **data warehouse global**: composto pelo **nível inferior** e por n_1 e n_2;
- **fragmentação**: baseada na dimensão filial
- **replicação**: do nível n_2
- **organização dos *sites***: utilização de 3 *sites*, sendo que:
  - ***site* 1**: composto pelo **nível inferior** e pelo nível n_1 fragmentados com relação à dimensão filial (ch_filial = 1) e por uma réplica de n_2
  - ***site* 2**: composto pelo **nível inferior** e pelo nível n_1 fragmentados com relação à dimensão filial (ch_filial = 2) e por uma réplica de n_2
  - ***site* 3**: composto pelo **data warehouse global**

A implementação do **login site** foi realizada de maneira a suportar as funcionalidades de autenticação de usuários (simplificada), oferecimento de interface gráfica, com tradução automática para a linguagem SQL suportada pelo SGBD SQL Server, e gerenciamento de consultas. A interface, de acordo com os requisitos apresentados na seção 4.2 (interface e tradução automática), permite que os usuários selecionem o tipo de relatório (dados relativos a vendas, lucros, produtos e promoções), a granularidade de tempo (diária, mensal ou anual) e as unidades geográficas envolvidas na consulta (filial 1, filial 2 ou ambas filiais). Por fim, foram desenvolvidos módulos, baseados nas especificações da seção 4.2 (redirecionamento de consultas) para: (a) seleção dos *sites* que poderiam responder a uma dada consulta, (b) escolha do conjunto de *sites* de (a) que melhor atende ao processamento da consulta e (c) redirecionamento da consulta para o conjunto escolhido. Tais módulos garantiram a transparência de localização proposta pela arquitetura.

6. Conclusões e Extensões

Este artigo descreveu uma arquitetura de replicação/fragmentação para acesso distribuído a *data warehouse* via Web, considerando-se um grande número de usuários. A arquitetura foi apresentada sob três enfoques: distribuição dos dados do *data warehouse*, forma na qual consultas submetidas ao *data warehouse* são redirecionadas através do **login site** e solução para o acesso aos dados. O artigo também apresentou um estudo de caso baseado na arquitetura proposta para uma cadeia de supermercados.

A arquitetura proposta neste artigo tem como objetivos aumentar a disponibilidade dos dados do *data warehouse*, aumentar a disponibilidade de acesso ao *data warehouse*, garantir transparência de localização, aumentar o desempenho no processamento de consultas e prover portabilidade.

O aumento da disponibilidade dos dados é obtido através da presença de um **data warehouse global** e de sua replicação e fragmentação em vários *sites*, enquanto que o aumento da disponibilidade de acesso é obtido pela possibilidade de duplicação do **login site** e pela presença de **wrappers** locais. O **login site** também é responsável pelo redirecionamento de consultas centralizadas e pelo gerenciamento de consultas distribuídas de forma com que os usuários não se preocupem com a localização dos dados, além do fato do *site* a ser acessado estar disponível ou não. Já a distribuição dos dados do *data warehouse* em vários *sites* contribui para o aumento de desempenho no processamento de consultas, devido à execução: (1) de um número menor de transações em cada *site*, quando comparado ao número total de transações que são submetidas ao sistema e (2) paralela de transações, especialmente quando estas acessam as mesmas porções de dados. Por fim, a garantia de portabilidade está relacionada à utilização da arquitetura *three-tier*, que permite independência da aplicação no acesso ao banco de dados.

Em adição às características acima apresentadas, a arquitetura provê facilidade de utilização através de uma interface gráfica, via Web, que permite que usuários finais, típicos de sistemas de tomada de decisão, selecionem suas consultas independentemente de uma linguagem de consulta de banco de dados.

Uma primeira extensão a este trabalho consiste na análise dos efeitos provocados na arquitetura caso o carregamento adicional de informações seja realizado *on-line*. Outras extensões, de um ponto de vista mais prático, se referem ao oferecimento de um módulo que permita que consultas possam ser efetuadas diretamente através da submissão de comandos SQL, à criação de uma ferramenta de monitoração que mantenha informações estatísticas sobre o redirecionamento de consultas e à inserção de agentes inteligentes no suporte à função de redirecionamento de consultas.

Referências Bibliográficas

[BPT97] Baralis, E., Paraboschi, S., Teniente, E. Materialized View Selection in a Multidimensional Database. In *Proc. 23rd VLDB Conference*, pp. 156-165, Athens, Greece, August 1997.

[CCS93] Codd, E.F., Codd, S.B., Salley,C.T. *Providing OLAP (On-line Analytical Processing) to User-Analysts: An IT Mandate*. White paper. 15 pp. Available at URL http://www.arborsoft.com/papers.

[CD97] Chaudhuri, S., Dayal, U. An Overview of Data Warehousing and OLAP Technology. *SIGMOD Record*, 26(1):65-74, 1997.

[CR97] Campos, M.L., Rocha Filho, A.V. Data Warehouse. In *Anais XVI JAI – XVII Congresso da SBC*, pp.221-261, Brasília, DF, Agosto 1997.

[CT98] Cabibbo, L., Torlone, R. A Logical Approach to Multidimensional Databases. In *Proc. 6th EDBT*, pp.183-197, Valencia, Spain, March 1998.

[EN94] Elmasri, R., Navathe, S.B. *Fundamentals of Database Systems*. Addison-Wesley Publishing Company, USA, 1994. 873 pp.

[FDLS97] Fernandez, M.F., Florescu, D., Levy, A.Y., Suciu, D. A Query Language for a Web-site Management System. *SIGMOD Record*, 26(3):4-11, 1997.

[FGLM+98] Fankhauser, P., Gardarin, G., Lopez, M., Munoz, J., Tomasic, A. Experiences in Federated Databases: From IRO-DB to MIRO-Web. In *Proc. 24th VLDB Conference*, pp. 655-658, NY, USA, 1998.

[GBLP95] Gray, J. Bosworth, A., Layman, A., Pirahesh, H. Data Cube: A Relational Aggregation Operator Generalizing Group-By, Cross-Tab, and Sub-Totals. Technical report MSR-TR-95-22, Advanced Technology Division, Microsoft Corporation, Redmond, USA, 1995. 8 pp.

[GL97] Gyssens, M., Lakshmanan, L.V.S. A Foundation for Multi-Dimensional Databases. In *Proc. 23rd VLDB Conference*, pp.106-115, Athens, Greece, August 1997.

[GMR98] Golfarelli, M., Maio, D., Rizzi, S. Conceptual Design of Data Warehouses form E/R Schemes. In *Proc. 31st HICSS*, pp.334-343, Kona, Hawaii, January 1998.

[HGMW+95] Hammer, J., Garcia-Molina, H., Widom, J., Labio, W., Zhuge, Y. The Stanford Data Warehousing Project. *IEEE Data Engineering Bulletin*, 18(2):41-48, 1995.

[Huy97] Huyn, N. Multiple-View Self-Maintenance in Data Warehousing Environments. In *Proc. 23rd VLDB Conference*, pp.16-25, Athens, Greece, August 1997.

[IBM99] IBM Visual Warehouse. Available at URL http://www.software.ibm.com/datamart.

[Inm96] Inmon, W.H. *Building the Data Warehouse*. John Wiley & Sons, Inc, USA, 1996. 401pp.

[Kim96] Kimball, R. *The Data Warehouse Toolkit*. John Wiley & Sons, Inc, USA, 1996. 388 pp.

[MQM97] Mumick, I.S., Quass, D., Mumick, B.S. Maintenance of Data Cubes and Summary Tables in a Warehouse. In *Proc. SIGMOD Conference on Management of Data*, pp.100-111, Tucson, Arizona, 1997.

[MZ98] Milo, T., Zohar, S. Schema-Based Data Translation. In *Proc. WebDW*, Valencia, Spain, March 1998. Available at URL http://www.dia.uniroma3.it/webdw98/papers.html.

[Ora99a] Oracle Warehouse. Available at URL http://www.oracle.com/datawarehouse.

[Ora99b] Oracle Express OLAP Technology. Available at URL http://www.oracle.com/olap.

[Red99] Red Brick Systems Inc. Available at URL http://www.redbrick.com.

[Ree97] Reese, G. *Database Programming with JDBC and JAVA*. O'Reilly & Associates, Inc, USA, 1997. 224 pp.

[Rou97] Roussopoulos, N. Materialized Views and Data Warehouses. In *Proc. 4th KRDB Workshop*, Athens, Greece, August 1997. Available at URL http://www.cs.umd.edu/~nick/papers/My_Views_on_Views.html.

[RS97] Ross, K.A., Srivastava, D. Fast Computation of Sparce Datacubes. In *Proc. 23rd VLDB Conference*, pp.116-125, Athens, Greece, August 1997.

[SC98] Siméon, J., Cluet, S. Using YAT to Build a Web Server. In *Proc. WebDW*, Valencia, Spain, March 1998. Available at URL http://www.dia.uniroma3.it/webdw98/papers.html.

[SDN98] Shukla, A., Deshpande, P.M., Naughton, J.F. Materialized View Selection for Multidimensional Datasets. In *Proc. 24th VLDB Conference*, pp.488-499, New York, USA, 1998.

[Sho97] Shoshani, A. OLAP and Statistical Databases: Similarities and Differences. In *Proc. ACM PODS*, pp.185-196, Tucson, Arizona, May 1997.

[SSU96] Silberschatz, A., Stonebraker, M., Ullman, J., editors. Database Research: Achievements and Opportunities Into the 21st Century. *Journal of the Brazilian Computer Society*, 3(2):5-10, 1996.

[WB97] Wu, M-C, Buchmann, A.P. Research Issues in Data Warehousing. In *BTW'97*, pp.61-82, Ulm, 1997.

[WGL+96] Wiener, J.L., Gupta, H., Labio, W.J., Zhuge, Y., Garcia-Molina, H., Widom, J. A System Prototype for Warehouse View Maintenance. *In Proc. of ACM Workshop on Materialized Views: Techniques and Applications*, pp.26-33, Montreal, Canada, 1996.

[Wid95] Widom, J. Research Problems in Data Warehousing. In *Proc. 4th International CIKM*, pp.25-30, Baltimore, Maryland, December, 1995.

[Wie97] Wiener, J.L. Data Warehousing: What is it? & related Stanford DB research? An overview talk given in the Stanford DB Seminar series, Stanford University, USA, 1997. 55 pp.

[YKL97] Yang, J., Karlapalem, K., Li, Q. Algorithms for Materialized View Design in Data Warehousing Environment. In *Proc. 23rd VLDB Conference*, pp.136-145, Athens, Greece, August 1997.