



Universidade Federal de Pernambuco
Graduação em Ciência da Computação
Centro de Informática



ANÁLISE COMPARATIVA DE ARQUITETURAS DE DISTRIBUIÇÃO DE DATA WAREHOUSE

TRABALHO DE GRADUAÇÃO

Aluno: Álvaro Alencar Barbosa Palitot (aabp@cin.ufpe.br)

Orientador: Fernando da Fonseca de Souza (fdfd@cin.ufpe.br)

Recife, 04 de Abril de 2007

Universidade Federal de Pernambuco
Graduação em Ciência da Computação
Centro de Informática

2006.2

Álvaro Alencar Barbosa Palitot

ANÁLISE COMPARATIVA DE ARQUITETURAS DE DISTRIBUIÇÃO DE DATA WAREHOUSE

ESTE TRABALHO FOI APRESENTADO À GRADUAÇÃO EM
CIÊNCIA DA COMPUTAÇÃO DO CENTRO DE INFORMÁTICA DA
UNIVERSIDADE FEDERAL DE PERNAMBUCO COMO REQUISITO
PARCIAL PARA OBTENÇÃO DO GRAU DE BACHAREL EM
CIÊNCIA DA COMPUTAÇÃO.

ORIENTADOR: PROF. DR. FERNADO DA FONSECA DE SOUZA

Recife, 04 de Abril de 2007

Dedico,

Aos meus pais

Ademar e Socorro

Aos meus avós paternos

Adauto e Francisca

Aos meus avós maternos

Geraldo e Zuleica

Agradecimentos

Primeiramente, agradeço a Deus por ter me dado o dom da vida.

Aos meus pais, Ademar e Socorro, pelo constante incentivo, carinho, compreensão, dedicação, amor e esforço para que eu pudesse chegar até aqui.

Aos meus avós paternos, Adauto e Francisca, e os meus avós maternos, Geraldo e Zuleica, pela força que sempre me deram durante toda a minha vida. Em especial, para o meu avô Adauto, que sempre acreditou no meu potencial, e apesar de não estar presente fisicamente hoje, me ilumina todos os dias.

Aos meus irmãos, André e Daniela, pela amizade e compreensão.

A minha namorada, Luciana, pelo término do trabalho, em detrimento de vê-la.

Ao meu orientador, Fernando da Fonseca de Souza, pela amizade, incentivo, confiança e disponibilidade na elaboração do trabalho.

E, finalmente, a todos aqueles que contribuíram direta ou indiretamente para a realização deste trabalho.

Resumo

Um *data warehouse* consiste em uma coleção de dados orientada por assuntos, integrada, variante no tempo e não volátil que dá suporte à tomada de decisão pela alta gerência da empresa.

Em geral, o *data warehouse* representa uma única base de dados centralizada. Entretanto, dados extremamente centralizados podem resultar em perda de disponibilidade e queda de desempenho das consultas. Daí surge a necessidade de um ambiente de distribuição de *data warehouse*, tendo como vantagens sobre os ambientes centralizados: o aumento da disponibilidade dos dados, o aumento da disponibilidade de acesso aos dados e o aumento de desempenho no processamento de consultas OLAP.

Este trabalho analisa propostas de arquiteturas destes ambientes de *data warehousing* distribuído, em relação a um conjunto de critérios. Outra contribuição é a apresentação de recomendações para a construção de arquiteturas de *data warehousing* distribuído.

Palavras-chave:

Data Warehouse, Data Warehousing, Sistemas de Suporte à Decisão, Data Warehouse Distribuído.

Abstract

A *data warehouse* consists of an integrated, variant in time, non-volatile and subject-oriented data collection that supports senior management decision-making.

In general, the *data warehouse* represents a unique centered database. However, extremely centered data might result in availability loss and decrease of queries performance. Therefore, it is needed a *data warehouse* distribution environment. Its advantages over centered environments are: increase of data availability and data access availability; and increase of OLAP queries processing performance.

This work analyses distributed *data warehouse* architecture proposals according to some criteria. Another contribution is the presentation of recommendations for constructing distributed *data warehouse* architectures based on the analysis carried out in this work .

Keywords:

Data Warehouse, Data Warehousing, Decision Support System, Distributed Data Warehouse.

Sumário

1. Introdução	11
1.1 Motivação	11
1.2 Objetivo	12
1.3 Estrutura do Trabalho	12
2. Data Warehousing	13
2.1 Data Warehouse.....	13
2.2 Banco de Dados Convencional x Data Warehouse	14
2.3 Principais Características do Data Warehouse	14
2.4 Data Mart.....	15
2.5 OLAP	16
2.5.1 Servidor	16
2.5.2 Tipos de Sistemas OLAP	17
2.6 Arquitetura de Data Warehousing	19
2.6.1 Componente de Integração e Manutenção.....	20
2.6.2 Componente de Análise e Consulta.....	21
2.7 Abordagens para desenvolvimento do Data Warehousing.....	21
2.7.1 Abordagem <i>Top-Down</i>	22
2.7.2 Abordagem <i>Bottom-Up</i>	22
2.7.3 Abordagem Intermediária ou Corrente.....	23
3. Distribuição de Data Warehouse.....	24
3.1 Banco de Dados Distribuídos	24
3.2 Banco de Dados Distribuído x Data Warehouse Distribuído	25
3.3 Arquitetura de Data Warehouse Distribuído de Inmon	25
3.4 Arquitetura de Data Warehousing Distribuído de Moeller	27
3.4.1 Arquitetura de Data Warehousing Distribuído Homogêneo	27
3.4.2 Arquitetura de Data Warehousing Distribuído Heterogêneo	28
3.4.3 Arquitetura de Data Warehousing Distribuído com SGBD Distribuído Único ..	29
3.4.4 Abordagem para o Problema de Localização dos Dados	30
3.5 Arquitetura de Data Warehouse Distribuído Hierárquico de Zhou et al.	32
3.5.1 HDDW orientado a consultas OLAP.....	34
3.6 Arquitetura de Data Warehouse Distribuído de Ciferri (O Sistema WebD ² W)	35
3.6.1 Arquitetura.....	36
3.6.2 A importância do Data Warehouse Global.....	38
3.6.3 Componente de Integração e Manutenção.....	38
3.6.4 Componente de Manutenção do ambiente distribuído	38
3.6.5 Componente de Distribuição	39
3.6.5.1 Módulo Requisitos.....	39
3.6.5.2 Módulo Fragmentação.....	39
3.6.5.3 Módulo Alocação	40
3.6.5.4 Módulo Carga	40
3.6.6 Componente de Consulta do Ambiente Distribuído.....	40
3.6.7 Arquitetura do Sistema WebD ² W na Web	41

4. Análise Comparativa de Arquiteturas de Distribuição de Data Warehouse	42
4.1 Análise da Arquitetura de Data Warehouse Distribuído de Inmon	43
4.2 Análise das Arquiteturas de Data Warehousing Distribuído de Moeller	44
4.3 Análise da Arquitetura de Data Warehouse Distribuído Hierárquico de Zhou et al. .	46
4.4 Análise do Sistema WebD ² W de Ciferri	48
4.5 Benefícios e Desvantagens das arquiteturas abordadas.....	49
4.6 Recomendações para a construção de arquiteturas de data warehousing distribuído	51
5. Conclusões e Trabalhos Futuros	53
5.1 Trabalhos Futuros	53
Referências Bibliográficas:	55

Lista de Figuras

Figura 2.1 – Data Marts com dados referentes a um determinado assunto [DAL99]	15
Figura 2.2 – Arquitetura ROLAP [TVFR07]	17
Figura 2.3 – Arquitetura MOLAP [TVFR07]	18
Figura 2.4 – Arquitetura HOLAP [TVFR07]	18
Figura 2.5 – Resumo das características dos tipos de OLAP mais usados [TVFR07]	19
Figura 2.6 – Arquitetura Típica de um ambiente de Data Warehousing [CIFE02].....	19
Figura 2.7 – Abordagem Top-Down da arquitetura de Data Warehousing [TVFR07].....	22
Figura 2.8 – Abordagem Bottom-Up da arquitetura de Data Warehousing [TVFR07]	22
Figura 2.9 – Abordagem Corrente da arquitetura de Data Warehousing [TVFR07]	23
Figura 3.1 – Arquitetura básica de data warehouse distribuído de Inmon [CIFE02]	25
Figura 3.2 – Variação da arquitetura básica de data warehouse distribuído de Inmon [CIFE02]	26
Figura 3.3 – Arquitetura de Data Warehousing Distribuído Homogêneo de Moeller [CIFE02]	28
Figura 3.4 – Arquitetura de Data Warehousing Distribuído Heterogêneo de Moeller [CIFE02]	29
Figura 3.5 – Arquitetura de Data Warehousing Distribuído com SGBD distribuído único de Moeller [CIFE02]	30
Figura 3.6 – Esquema estrela da informação econômica do Data Warehouse.....	32
Figura 3.7 – DW Centralizado (Baseado em [AJLW98])	32
Figura 3.8 – DW Distribuído ou Federado (Baseado em [AJLW98]).....	32
Figura 3.9 – Data Warehouse Distribuído Hierárquico (Baseado em [ZZTH00]).....	33
Figura 3.10 - HDDW orientado a consultas OLAP (Baseado em [ZZTH00]).....	34
Figura 3.11 – Mecanismo OLAP do sistema HDDW (Baseado em [ZZTH00])	35
Figura 3.12 – Arquitetura básica do sistema WebD ² W [CIFE02].....	37

Lista de Quadros

Quadro 2.1 – Comparações entre o Banco de Dados Convencional e o Data Warehouse [INM96; BAR96; KIM96; ONE97].....	14
Quadro 2.2 – Diferenças entre DW e DM [NASC05].....	16
Quadro 3.1 – Comparações entre DW Centralizado, DW Distribuído de Wells et al. e HDDW (Baseado em [ZZTH00]).....	33
Quadro 4.1 – Funcionalidades oferecidas pela arquitetura de data warehouse distribuído de Inmon [INM02; CIFE02].....	43
Quadro 4.2 – Funcionalidades oferecidas pelas arquiteturas de data warehousing distribuído de Moeller [MOE01; CIFE02]	45
Quadro 4.3 – Funcionalidades oferecidas pela arquitetura de data warehouse distribuído Hierárquico de Zhou et al [ZZTH00]	47
Quadro 4.4 – Funcionalidades oferecidas pelo Sistema WebD ² W [CIFE02]	48

1. Introdução

A quantidade de informações e dados nas empresas tem crescido, criando uma enorme dificuldade para que os profissionais consigam tratar todo esse conteúdo disponível. Há, com isso, a necessidade de se filtrar informações de forma adequada, utilizando-se somente aquelas realmente relevantes ao modelo de negócio das empresas. O modelo que tenta simplificar os dados disponíveis e direcioná-los para os gestores está vinculado a uma tecnologia denominada de *data warehousing* [TERRA07].

Um ambiente de *data warehousing* transforma dados operacionais em informação voltada à tomada de decisão estratégica. Para tanto, oferece um conjunto de funcionalidades que possibilita, por um lado, a extração, tradução, filtragem, integração e armazenamento no *Data Warehouse* (DW) dos dados oriundos de provedores de informação autônomos, distribuídos e heterogêneos. Por outro lado, este conjunto de funcionalidades também permite que usuários típicos de Sistemas de Suporte à Decisão (SSD), por exemplo, analistas, executivos, gerentes e administradores, manipulem com flexibilidade e eficiência os dados pré-armazenados no *data warehouse*, através de visões multidimensionais destes dados [CHAU97].

O DW, principal componente do ambiente de *data warehousing*, consiste em um banco de dados especial, organizado para armazenar uma coleção de dados integrados, orientados por assunto, variáveis com o tempo e não voláteis, usados para dar suporte ao processo gerencial de tomada de decisão [INM02].

Normalmente, o *data warehouse* representa uma única base de dados centralizada. Distribuir os dados armazenados nessa base de dados levando-se em consideração as características particulares de aplicações de *data warehousing* e as necessidades dos usuários típicos de SSD representa, portanto, uma área de pesquisa muito importante a ser explorada [CIFE02].

Isto motivou pesquisadores ao desenvolvimento de propostas de arquitetura de *data warehousing* distribuído que solucionem os desafios introduzidos por esta distribuição, gerenciando os diversos *data warehouses* acoplados aos diferentes computadores ligados entre si por redes de comunicação e permitido que as informações dessas bases de dados sejam armazenadas e recuperadas rapidamente.

1.1 Motivação

Atualmente, a descentralização das funções de negócios e a distribuição das corporações em regiões geográficas distintas motivam a construção de arquiteturas de *data warehousing* distribuído.

As vantagens proporcionadas pela distribuição de *data warehouses*, como as listadas abaixo, são algumas das motivações para o desenvolvimento desse trabalho:

- Melhoria da confiabilidade e disponibilidade dos dados, devido aos dados poderem ser acessados em diferentes *sites*, aumentando o processamento distribuído;
- Suporte a um maior número de usuários;
- Melhoria no desempenho das consultas OLAP, devido ao processamento distribuído das consultas;
- Expansão e manutenção mais fáceis, devido ao fraco acoplamento entre os *sites*; e

- Transparência de fragmentação, alocação e replicação, proporcionando uma visão para o usuário como se fosse um sistema centralizado, mas com as vantagens da distribuição.

1.2 Objetivo

É em razão de todas estas motivações, que este trabalho tem como principais objetivos: analisar soluções para ambientes de *data warehouse* distribuído, fazer um estudo comparativo entre as mesmas, indicando vantagens e desvantagens destas arquiteturas e elaborar recomendações para a construção de arquiteturas de *data warehouse* distribuído.

1.3 Estrutura do Trabalho

Além deste capítulo introdutório, o trabalho é composto por mais cinco capítulos. O capítulo 2 tem por objetivo uma introdução a *data warehousing*, discutindo os principais conceitos, componentes e funcionalidades desses ambientes. O capítulo 3 faz uma breve comparação entre banco de dados distribuídos e *data warehouse* distribuídos, além de abordar as quatro arquiteturas existentes atualmente de *data warehousing* distribuído, propostas respectivamente feitas por Imon [INM02], Moeller [MOE01], Zhou et al. [ZZTH00] e Ciferri [CIFE02]. O capítulo 4 faz uma análise comparativa em relação a alguns critérios pré-estabelecidos no próprio capítulo entre as arquiteturas de *data warehousing* distribuído, além de especificar vantagens e desvantagens de cada uma destas arquiteturas e elaborar recomendações para a construção de arquiteturas voltadas para a distribuição de *data warehouse*. O capítulo 5 apresenta a conclusão e propostas para trabalhos futuros.

2. Data Warehousing

Atualmente, há uma grande quantidade de dados nas empresas, porém faltam informações significativas. Isto ocorre devido à extensa utilização de sistemas de informação específicos, contribuindo para a pouca integração dos dados e a existência de ambientes não muito adequados para o tratamento analítico dos dados.

O ambiente de *data warehousing* atende a esta necessidade, promovendo a integração dos dados dos sistemas de informação espalhados pela organização e gerando um ambiente de consultas com informações significativas, que permitem à alta gerência, análises que ajudam no processo de tomada de decisão da empresa [FFRC07].

Antes de definir o termo *data warehouse*, é preciso fazer uma distinção entre *data warehouse* e *data warehousing*, embora a maior parte da literatura trate os dois termos de maneira idêntica. Sempre que ocorrerem referências ao termo *data warehouse* neste trabalho, trata-se de um banco de dados especial que oferece o suporte aos usuários no processo de tomada de decisão. Já as referências a *data warehousing* devem ser tratadas como todo o ambiente de *data warehouse*, que engloba o próprio *data warehouse*, suas arquiteturas, algoritmos e ferramentas, usuários, componentes para realização de consulta, entre outros [CIFE02].

Através da arquitetura de um ambiente de *data warehousing* é possível identificar os componentes que participam no ambiente, o relacionamento que existe entre estes componentes e as funcionalidades de cada um. Separar estes dois conceitos essenciais é importante, a fim de evitar problemas relacionados ao uso destes termos.

2.1 Data Warehouse

Inicialmente, serão analisadas algumas definições para *data warehouse* elaboradas por autores da área.

Segundo Inmon [INM96], o idealizador do conceito, *Data Warehouse* “é uma coleção de dados integrados, orientados por assunto, variáveis com o tempo e não voláteis, usados para dar suporte ao processo gerencial de tomada de decisão”.

Para Barquini [BAR96], *Data Warehouse* “é uma coleção de técnicas e tecnologias que juntas disponibilizam um enfoque pragmático e sistemático para tratar com o problema do usuário final que precisa acessar informações que estão distribuídas em vários sistemas da organização”.

Kimball [KIM96] define que “é um conjunto de ferramentas e técnicas de projeto, que quando aplicadas às necessidades específicas dos usuários e aos bancos de dados específicos permitirá que planejem e construam um *data warehouse*”.

Com base nestas definições, pode-se dizer, de forma resumida, que o *data warehouse* é um banco de dados especial integrado, que serve para dar suporte às altas gerências das empresas para análise de tendências históricas dos seus produtos, funcionários e clientes, aumentando, com isso, a competitividade da mesma.

No DW os dados são extraídos de múltiplas fontes de dados internas ou externas à empresa e depois são integrados. Os *data warehouses* são construídos para que tais dados possam ser armazenados e acessados de forma que não sejam limitados por tabelas e linhas estritamente relacionais.

2.2 Banco de Dados Convencional x Data Warehouse

É de fundamental importância para este trabalho explicitar as diferenças existentes entre essas duas metodologias de armazenamento de dados, para permitir melhor compreensão de seus objetivos. A principal diferença se caracteriza pelo *data warehouse* ser um ambiente informacional, enquanto que os bancos de dados convencionais das organizações, como os localizados em suas filiais, serem ambientes operacionais.

De acordo com Inmon [INM96], em um ambiente operacional, manipula-se um volume grande de transações que geralmente são simples, pequenas e acessam poucos registros por vez. Já no ambiente informacional, manipula-se um baixo volume de transações que são longas, complexas e acessam muitos registros, necessitando muitas vezes realizar funções de junção e agregação.

A partir destas informações e de outras [INM96; BAR96; KIM96; ONE97], algumas diferenças existentes entre o banco de dados convencional e o *data warehouse* já podem ser verificadas. O Quadro 2.1 destaca estas diferenças:

QUADRO 2.1 – COMPARAÇÕES ENTRE O BANCO DE DADOS CONVENCIONAL E O DATA WAREHOUSE [INM96; BAR96; KIM96; ONE97]

Características	Banco de Dados Convencional	Data Warehouse
Objetivo	<i>Operações diárias do negócio</i>	<i>Analisar o negócio</i>
Usuários	<i>Funcionários</i>	<i>Alta Administração</i>
Tipo de Processamento	<i>OLTP</i>	<i>OLAP</i>
Unidade de Trabalho	<i>Inclusão, alteração, exclusão e consulta</i>	<i>Carga e Consulta</i>
Organização dos dados	<i>Orientado a Aplicações</i>	<i>Orientado a Assunto</i>
Condições dos Dados	<i>Dados Operacionais</i>	<i>Dados Analíticos</i>
Interação do Usuário	<i>Somente pré-definida</i>	<i>Pré-definida e ad-hoc</i>
Volume	<i>Megabytes – Gibabytes</i>	<i>Gigabytes – Terabytes</i>
Histórico	<i>60 a 90 dias</i>	<i>5 a 10 anos</i>
Granularidade	<i>Detalhados</i>	<i>Detalhados e resumidos</i>
Redundância	<i>Não ocorre</i>	<i>Ocorre</i>
Acesso a registros	<i>Dezenas</i>	<i>Milhares</i>
Atualização	<i>Contínua (tempo real)</i>	<i>Periódica (em batch)</i>
Número de índices	<i>Poucos/ Simples</i>	<i>Muitos/Complexos</i>
Intenção de índices	<i>Localizar um registro</i>	<i>Aperfeiçoar consultas</i>
Manutenção desejada	<i>Mínima</i>	<i>Constante</i>

2.3 Principais Características do Data Warehouse

A partir da definição de DW feita por Inmon [INM96], vista anteriormente, é possível encontrar as principais características de *data warehouse*, que são:

- Integrado - A integração dos dados ocorre quando os dados são passados do ambiente operacional, através de diversas fontes de dados, para o ambiente de DW. Todo dado trazido dos sistemas operacionais para o ambiente de DW é,

anteriormente, consolidado, de forma que passe a ter um único significado. É durante a integração, que os dados são convertidos para um estado uniforme;

- Orientado por assuntos - Os dados em um *data warehouse* são organizados de modo a facilitar a análise dos dados, para isso o DW contém informações orientada a assuntos importantes para o negócio da empresa e não por aplicação, como em bancos de dados operacionais;
- Variante no Tempo - Os dados não são atualizáveis, ou seja, são relativos a um determinado instante de tempo, o que proporciona o armazenamento do histórico dos dados; e
- Não volátil – Significa dizer que o *data warehouse* permite apenas a carga inicial dos dados e consultas a estes dados. Ou seja, após serem integrados, transformados e incluídos, os dados não podem ser alterados.

2.4 Data Mart

Data Mart (DM) corresponde a um subconjunto lógico do *data warehouse*, geralmente visto como um *data warehouse* setorial ou departamental. Entretanto, os dados armazenados nos *data marts* têm as mesmas características que os dados do DW, vistas anteriormente. A Figura 2.1 mostra três DM, cada um referente a um determinado assunto (Vendas, Compras e Estoque) [DAL99].

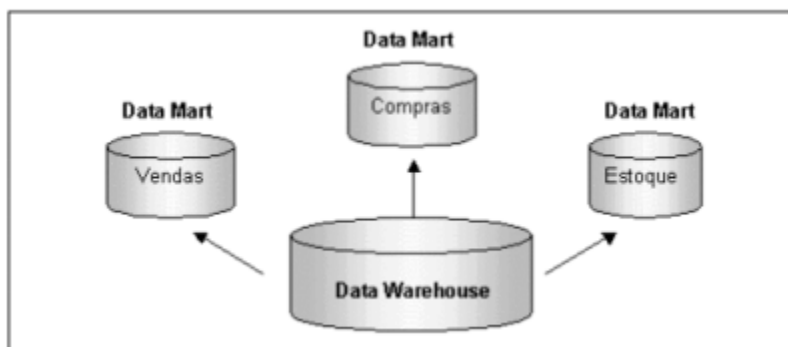


FIGURA 2.1 – DATA MARTS COM DADOS REFERENTES A UM DETERMINADO ASSUNTO [DAL99]

Segundo Inmon [INM96], os DM apresentam as seguintes características:

- São especificados para atender a uma área ou conjunto de áreas de interesse;
- Empregam normalmente um esquema estrela no projeto de banco de dados. Esta modelagem é elaborada com base nas exigências dos usuários finais;
- Contêm uma quantidade razoável de informações históricas, normalmente, menor que o volume histórico do DW;
- Apresentam uma granularidade, normalmente, maior que a do DW. Esta granularidade tem o propósito de atender às necessidades do usuário final; e
- Apresentam um armazenamento dos dados altamente indexado

Numa visão comparativa do *data warehouse* com o *data mart*, considerando os critérios: escopo, integração, tempo, agregação, análise e dados voláteis, percebe-se que a diferença está no escopo, pois enquanto o DW é feito para atender uma empresa como um todo, o DM é criado para atender um subconjunto da empresa. Deve-se observar que atender a um subconjunto da empresa pode significar reunir dados de outros setores, já que, na prática, raramente, um único setor possui ou gera toda informação que precisa.

O Quadro 2.2 apresenta mais diferenças entre o DW e o DM.

QUADRO 2.2 – DIFERENÇAS ENTRE DW E DM [NASC05]

Data Warehouse	Data Mart
Corporativo	<i>Departamental</i>
Granularidade em baixo nível. Dados bem detalhados	<i>Granularidade em alto nível</i>
Estrutura normalizada (com tratamento)	<i>Emprega o esquema estrela como estrutura de dados</i>
Grande volume de histórico de dados	<i>Não armazena grande volume de dados históricos</i>
Emprega tecnologia orientada ao armazenamento de grandes volumes de dados	<i>Emprega tecnologia multidimensional excelente para acesso e análise</i>
Modelagem de dados com o propósito de atender à corporação	<i>Modelagem de dados com o objetivo de atender a um usuário final</i>
Levemente indexado	<i>Altamente indexado</i>

2.5 OLAP

A origem do nome OLAP baseia-se no acrônimo para *Online Analytical Processing*, o qual significa em português: Processamento Analítico em Tempo Real. O termo Online significa que as operações da ferramenta deveriam ter uma resposta imediata, ou seja, em tempo real. Já a palavra Analytical demonstra o uso de teorias analíticas para tornar as buscas possíveis. O vocábulo Processing reforça a característica de intenso processamento quando as consultas em uma grande quantidade de dados são realizadas [CHAU97].

O processamento analítico em tempo real é a tecnologia de software *front-end* que capacita os analistas, gerentes e executivos a obter discernimento nos dados através de um acesso rápido, consistente e interativo. Esse discernimento pode ser realizado para uma larga variedade de possibilidades de visões da informação que vêm a ser transformadas a partir de simples dados para refletir o real dimensionamento da corporação como entendido pelo usuário.

2.5.1 Servidor

O servidor OLAP é o principal componente físico da arquitetura OLAP. Ele é o responsável por receber as requisições de consultas, processar esses pedidos e retornar o relatório. Para realizar essas tarefas, ele necessita de alta capacidade de processamento e suporte a multiusuário. O servidor OLAP se localiza entre o cliente e o SGBD [CHAU97].

Devido ao pré-processamento dos dados da base OLAP, o servidor opera em

estruturas multidimensionais e acessa os dados por intersecções entre as dimensões.

Dada a importância das ferramentas OLAP, os principais sistemas de gerenciamento de bancos de dados possuem um servidor OLAP. Estes são otimizados para a recuperação rápida de dados.

2.5.2 Tipos de Sistemas OLAP

Com a demanda da recuperação rápida de dados em diferentes ambientes, os sistemas OLAP se especializaram. Os principais tipos de sistemas OLAP estão classificados a seguir:

- ROLAP (OLAP Relacional) é considerado o mais escalável. Utiliza banco de dados relacionais para análise dos fatos, mas manipulando os dados de forma multidimensional via SQL. Lida com fatos atômicos e sumarizados e permite o uso de várias dimensões. Entretanto, é difícil implementar eficientemente um pré-processamento para grandes volumes, e por isto, é frequentemente deixado de lado. A Figura 2.2 ilustra a arquitetura ROLAP;

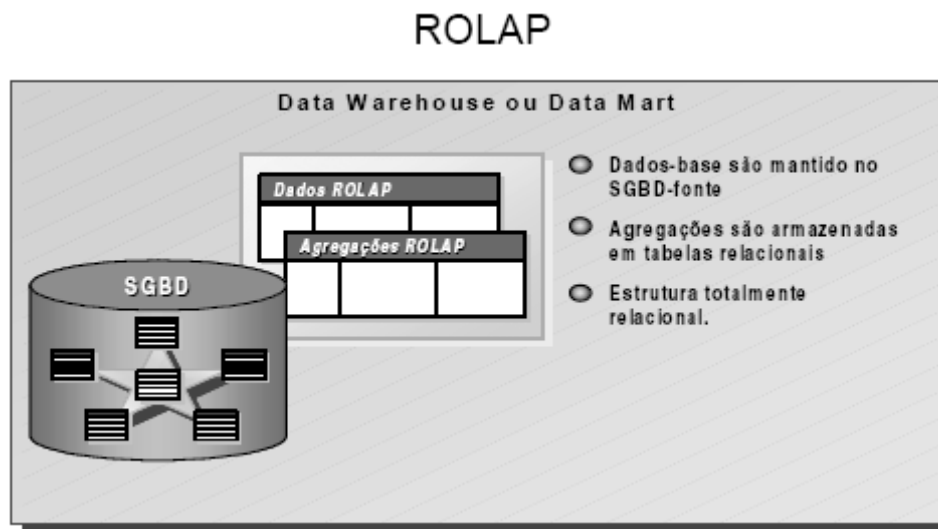


FIGURA 2.2 – ARQUITETURA ROLAP [TVFR07]

- MOLAP (OLAP Multidimensional) é o sistema que opera melhor com pequenas bases de dados. Utiliza banco de dados multidimensionais proprietários (MDDB) para manipular fatos agregados. Armazena de forma multidimensional para poder visualizar também de forma multidimensional. Devido ao tamanho das bases de dados, ele calcula rapidamente as agregações e repostas. Algumas desvantagens desse tipo de arquitetura OLAP são: o espaço para o seu armazenamento é limitado, não manipula fatos atômicos, não trabalha com

muitas dimensões e não gerencia um grande volume de fatos. A Figura 2.3 ilustra a arquitetura MOLAP;

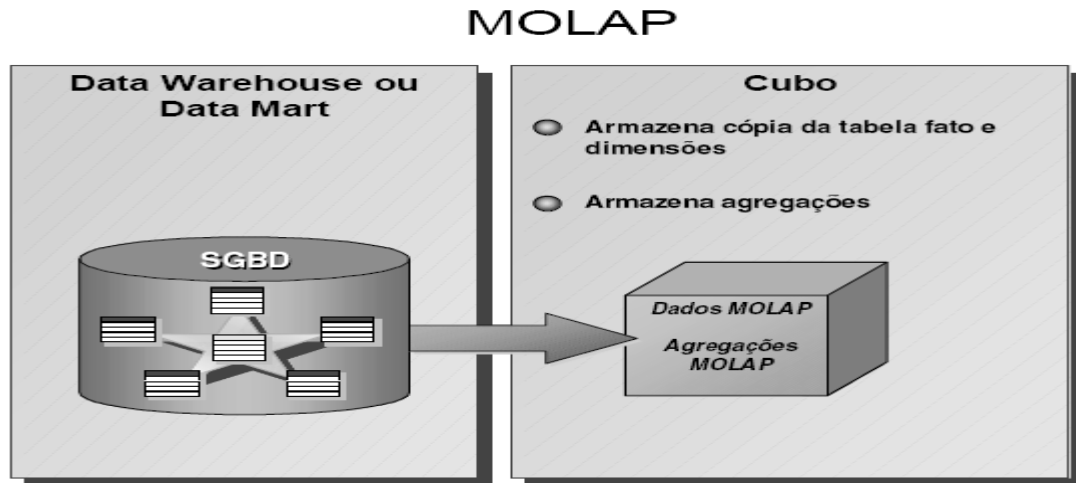


FIGURA 2.3 – ARQUITETURA MOLAP [TVFR07]

- HOLAP (OLAP Híbrido) é a solução intermediária entre o MOLAP e o ROLAP. Suporta manipulação de fatos atômicos e agregados, utilizando banco de dados multidimensionais para analisar fatos agregados. Utiliza SQL para manipular fatos atômicos. É o mais complexo para administrar e implementar. A Figura 2.4 ilustra a arquitetura HOLAP;

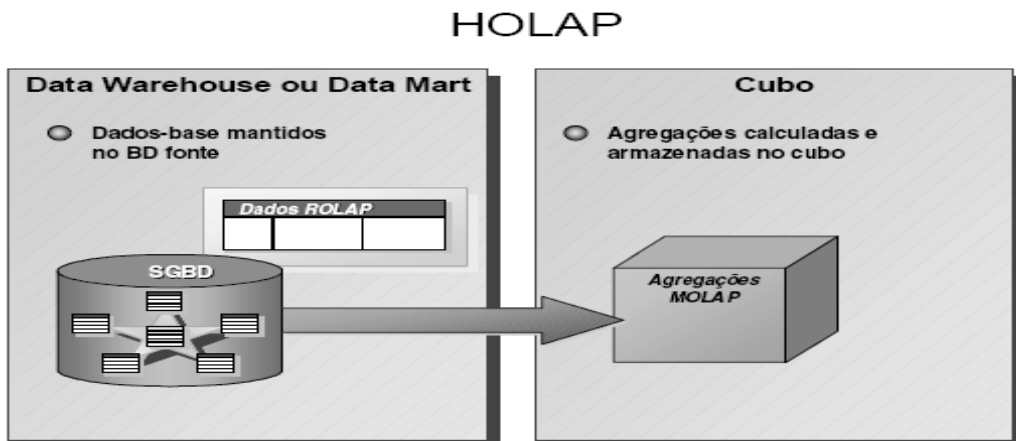


FIGURA 2.4 – ARQUITETURA HOLAP [TVFR07]

- WOLAP (Web OLAP) é o tipo para ser utilizado em sistemas web;
- DOLAP (Desktop OLAP) é uma especialização da ferramenta OLAP para ser usada no desktop; e
- RTOLAP (Real Time OLAP) é o sistema OLAP desenvolvido para ambientes em tempo real.

De todos estes tipos de OLAP, os mais usados atualmente são o ROLAP, MOLAP e o HOLAP. A Figura 2.5 mostra as principais características de cada uma destas arquiteturas.

Armazenamento	MOLAP	HOLAP	ROLAP
Dados de base	<i>Cubo</i>	<i>Tabela Relacional</i>	<i>Tabela Relacional</i>
Agregações	<i>Cubo</i>	<i>Cubo</i>	<i>Tabela Relacional</i>

Perspectiva do Cliente	MOLAP	HOLAP	ROLAP
Performance de consulta	<i>Imediato</i>	<i>Mais rápido</i>	<i>Rápido</i>
Consumo em disco	<i>Alto</i>	<i>Médio</i>	<i>Baixo</i>
Manutenção do cubo	<i>Alto</i>	<i>Médio</i>	<i>Baixo</i>

FIGURA 2.5 – RESUMO DAS CARACTERÍSTICAS DOS TIPOS DE OLAP MAIS USADOS [TVFR07]

2.6 Arquitetura de Data Warehousing

O objetivo de um ambiente de *data warehousing* é transformar o dado em conhecimento. A definição de uma arquitetura básica fornece a possibilidade de um melhor entendimento de todo o processo e das tecnologias que poderão ser utilizadas para otimizá-lo. Em razão disto, tem-se na Figura 2.6, uma imagem de uma arquitetura básica de um ambiente de *data warehousing* proposta por Ciferri [CIFE02].

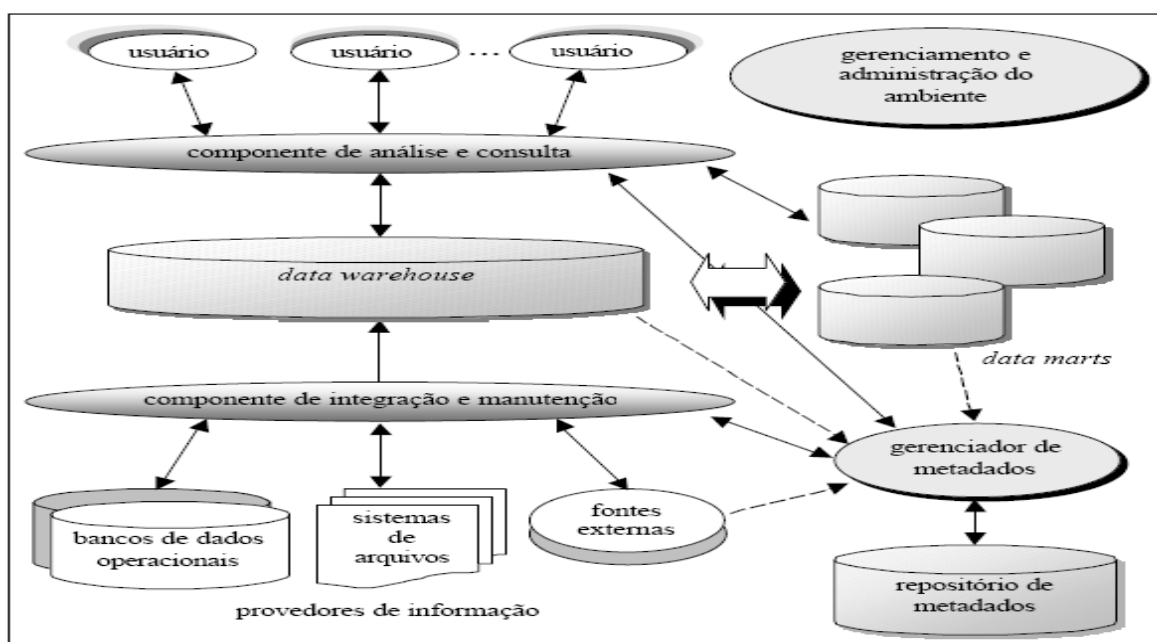


FIGURA 2.6 – ARQUITETURA TÍPICA DE UM AMBIENTE DE DATA WAREHOUSING [CIFE02]

Um estudo detalhado de cada componente da arquitetura da Figura 2.6, permite compreender como o ambiente de *data warehousing* faz para armazenar, integrar, comunicar, processar e apresentar os dados que os usuários utilizarão em suas decisões.

Os provedores de informação são compostos por dados operacionais, que podem estar presentes em várias formas de armazenamento de dados, como por exemplo: banco de dados operacionais, sistemas de arquivos e fontes externas.

Um ambiente de *data warehousing* pode utilizar tanto o *data warehouse* principal, quanto porções de dados do mesmo, ou seja, um conjunto de *data marts* que representam os fragmentos ou réplicas do *data warehouse* principal para o armazenamento dos dados. É através do componente de análise e consulta que as consultas dos usuários de SSD são submetidas e redirecionadas ao *data warehouse* principal ou aos *data marts* [NASC05].

O repositório de metadados armazena as informações estruturais e semânticas dos provedores de informação e do *data warehouse*, além de quaisquer outros dados importantes para o ambiente, enquanto que o módulo de gerenciamento de metadados manipula essas informações.

A arquitetura contém ferramentas para o gerenciamento e administração do ambiente, que são responsáveis pelo monitoramento do sistema, realizando importantes tarefas, tais como: o gerenciamento de segurança, testes de qualidade dos dados, *backup* dos dados e o gerenciamento e a atualização dos metadados, além de auditoria e relato da utilização do *data warehouse* [CIFE02].

Nos subtópicos a seguir, serão descritos detalhes das funcionalidades dos componentes de integração e manutenção e análise e consulta.

2.6.1 Componente de Integração e Manutenção

O componente de integração é responsável por proporcionar uma visão integrada de alta qualidade dos dados no *data warehouse*, sem duplicatas ou inconsistências. O componente de integração e manutenção oferece as funcionalidades de carregamento dos dados dos provedores de informação no *data warehouse*, atualização periódica desta base de dados e expiração de seus dados.

O carregamento dos dados dos provedores de informação consiste em operações de extração, tradução, limpeza, integração e carga dos dados. As ferramentas que auxiliam este processo são chamadas de ferramentas ETL, abordadas anteriormente neste trabalho.

A operação de extração é responsável por acessar as diversas fontes e extrair os dados que sejam úteis para o sistema de suporte à decisão. Depois de escolher os dados relevantes, os mesmos possuem muita inconsistência, e para isso precisam ser transformados. A operação de transformação engloba os processos de tradução, limpeza e integração de dados. O processo de tradução consiste na conversão dos dados das fontes para um formato padrão adotado pelo ambiente de *data warehousing*. O processo de limpeza transforma dados sujos em dados com qualidade, através de regras de negócio a eles associadas.

Segundo Ciferri [CIFE02], “o processo de integração depende da identificação de similaridades e de diferenças existentes entre os dados das fontes que foram previamente traduzidos, além da identificação de conjuntos destes dados que, apesar de serem distintos entre si, são relacionados por alguma propriedade semântica. Estas similaridades e diferenças devem ser detectadas tanto em nível de esquema quanto em nível de instância”.

A operação de carga consiste no processo de armazenamento dos dados integrados no *data warehouse*, após os processos descritos acima serem realizados. Além disto, durante esta operação são realizados processamentos adicionais, tais como: verificação de integridade, a ordenação dos dados, a geração de agregações, a construção de índices e a condensação dos dados, buscando diminuir o volume dos dados armazenados.

O processo de atualização dos dados serve para manter os dados integrados consistentes no *data warehouse*, em relação às fontes de dados. A periodicidade da manutenção do *data warehouse* depende da necessidade da alta gerência da empresa em ter os dados consistentes no *data warehouse* em relação às fontes de dados. Caso as consultas OLAP requeiram dados correntes, é necessário que toda alteração feita nos provedores de informação, sejam atualizadas imediatamente para o *data warehouse*.

O administrador do ambiente de *data warehousing* é quem estabelece a frequência com que estas alterações devem se propagar, podendo cada provedor de informação ter frequências diferentes. Geralmente, a frequência de manutenção é diária e ocorre no período em que a empresa efetua poucas ou nenhuma transações operacionais.

Recomputação e atualização são duas técnicas utilizadas para atualizar o *data warehouse*. Na recomputação, os dados no *data warehouse* são descartados e novamente carregados a partir dos provedores de informação operacionais. Já na atualização incremental, apenas as alterações feitas dos provedores de informação operacionais, desde a última atualização, são propagadas no *data warehouse* [CIFE02].

O processo de expiração dos dados serve para diminuir o volume de dados armazenados no *data warehouse*, que chega a ter terabytes de tamanho, dificultando em razão disso, a rapidez no processamento de consultas. Para que o volume diminua é preciso que o processo de expiração remova os dados do *data warehouse* que estejam “velhos” (atingam certo limite de tempo), quando não são mais relevantes, ou quando o espaço de armazenamento é insuficiente para armazenar novos dados advindos das fontes.

2.6.2 Componente de Análise e Consulta

O componente de análise e consulta serve para garantir o acesso às informações armazenadas no *data warehouse* aos usuários de SSD e aos *softwares* que participam do ambiente de *data warehousing*. Oferece, com isso, funcionalidades relacionadas à consulta e à análise dos dados armazenados, incluindo a habilidade de se determinar a origem dos dados que estão sendo examinados [CIFE02].

Esta fase é realizada através de aplicativos que oferecem acesso aos dados das fontes, através de interfaces amigáveis. Estas consultas solicitadas pelos usuários sofrem um processo de otimização, para que sejam executadas de forma a obter uma melhor performance [COST05].

2.7 Abordagens para desenvolvimento do Data Warehousing

No desenvolvimento de ambientes de *data warehousing* podem ser utilizadas as abordagens *top-down*, *bottom-up* e *intermediária*. A escolha da abordagem é fator essencial na seleção da tecnologia apropriada para o desenvolvimento e implantação deste ambiente.

2.7.1 Abordagem *Top-Down*

Nesta abordagem o desenvolvimento é feito de cima para baixo, ou seja, primeiro deve-se construir o *data warehouse* corporativo, para depois carregar os dados nos diversos *data marts* existentes. A Figura 2.7 apresenta a abordagem *top-down* da arquitetura do ambiente de *data warehousing*.



FIGURA 2.7 – ABORDAGEM TOP-DOWN DA ARQUITETURA DE DATA WAREHOUSING [TVFR07]

Esta abordagem possibilita várias vantagens para ambientes de *data warehousing*, como as listadas abaixo:

- Facilidade de manutenção devido a todos os *data marts* serem originados a partir de um mesmo *data warehouse*; e
- Necessidade de apenas um único conjunto de aplicações para extração, limpeza e integração dos dados.

Porém, esta abordagem apresenta algumas desvantagens:

- Implementação mais demorada; e
- Alto Risco.

2.7.2 Abordagem *Bottom-Up*

O propósito desta abordagem é a construção de um *data warehouse* incremental, a partir do desenvolvimento de *Data Marts* independentes. A Figura 2.8 apresenta a abordagem *bottom-up* da arquitetura de *data warehousing*.



FIGURA 2.8 – ABORDAGEM BOTTOM-UP DA ARQUITETURA DE DATA WAREHOUSING [TVFR07]

A abordagem *bottom-up* apresenta algumas vantagens, são elas:

- Apresentação de resultados em pouco tempo;
- Rápido desenvolvimento; e
- Permite os principais assuntos da empresa sejam desenvolvidos inicialmente.

Pode-se destacar como desvantagens:

- Necessidade de um maior controle do negócio da empresa, a fim de evitar “ilhas de dados” que dificultam futuras integrações; e
- Utilização de esforços e recursos de várias equipes, devido a, normalmente, o desenvolvimento ser feito em paralelo.

2.7.3 Abordagem Intermediária ou Corrente

Esta arquitetura tem o propósito de integrar a abordagem *top-down* com a *bottom-up*, com o planejamento sendo feito *top-down* e a implementação *bottom-up*. Segundo Soares [SOAR98], “nesta abordagem efetua-se a modelagem dos dados do DW, sendo o passo seguinte a implementação dos *Data Marts* escolhidos por área de interesse. Cada *Data Mart* gerado a partir do modelo de dados do *data warehouse* é integrado no modelo físico. A principal vantagem desta abordagem é a garantia da consistência dos dados e a diminuição ou eliminação das “ilhas de dados”. Esta garantia é obtida em virtude do modelo de dados para os *Data Marts* serem únicos, possibilitando realizar o mapeamento e o controle dos dados”. A Figura 2.9 apresenta a abordagem *intermediária* ou *corrente* da arquitetura de *data warehousing*.



FIGURA 2.9 – ABORDAGEM CORRENTE DA ARQUITETURA DE DATA WAREHOUSING [TVFR07]

3. Distribuição de Data Warehouse

A maioria das organizações constrói e mantém um único *data warehouse* centralizado, isto é feito por várias razões [INM02]:

- Os dados em um *data warehouse* é integrado pela organização, e uma visão integrada dos dados é usada somente na sede da organização;
- A organização opera em um modelo centralizado de negócio;
- O volume dos dados em um *data warehouse* é tal que um único repositório de dados centralizado faz sentido;
- Complexidade de desenvolvimento de um ambiente distribuído;
- Maior Segurança; e
- Fácil Gerenciamento.

Em resumo, a política, a economia e a tecnologia favorecem muito o uso de um único *data warehouse* centralizado.

Entretanto, dados extremamente centralizados podem resultar em perda de disponibilidade e queda de desempenho das consultas. Daí surge a necessidade de um ambiente de distribuição de *data warehouse*, tendo como vantagens sobre os ambientes centralizados: o aumento da disponibilidade dos dados, o aumento da disponibilidade de acesso aos dados e o aumento de desempenho no processamento de consultas OLAP [NOA00].

3.1 Banco de Dados Distribuídos

Os banco de dados distribuídos trazem vantagens da computação distribuída para o domínio do gerenciamento de banco de dados. Um sistema de computação distribuída consiste em vários elementos de processamento, não necessariamente homogêneos, que são interconectados por uma rede de computadores e cooperam na execução de certas tarefas [ERNS04].

Os banco de dados distribuídos podem ser definidos como uma coleção de múltiplos bancos de dados logicamente inter-relacionados, distribuídos por uma rede de computadores. Abaixo são destacadas algumas vantagens na utilização de banco de dados distribuídos:

- Transparência de fragmentação, replicação e alocação;
- Melhoria na confiabilidade e disponibilidade;
- Melhoria de desempenho; e
- Expansão mais fácil;

De acordo com Elmasri e Navathe [ERNS04], a distribuição leva a uma maior complexidade no projeto e na implementação do sistema. Para obter as vantagens potenciais listadas anteriormente, o ambiente de banco de dados distribuídos deve ser capaz de prover algumas funções, além daquelas já presentes em ambientes centralizados, como por exemplo:

- Rastreamento dos dados;
- Processamento de consultas distribuídas;

- Gerenciamento de transações distribuídas;
- Gerenciamento de dados replicados;
- Recuperação de banco de dados distribuído;
- Segurança; e
- Gerenciamento do diretório (catálogo) distribuído.

3.2 Banco de Dados Distribuído x Data Warehouse Distribuído

O *data warehouse* nada mais é do que um banco de dados especial integrado, orientado por assunto, variável com o tempo e não volátil, usado para dar suporte ao processo gerencial de tomada de decisão. Por isso, as contribuições obtidas pelos trabalhos de pesquisa em sistemas de banco de dados distribuídos podem ser utilizadas como base para o desenvolvimento de ambientes de *data warehousing* distribuídos.

Porém, esses trabalhos devem ser estendidos de forma a enfatizar aspectos importantes dos ambientes de *data warehousing* distribuído, tais como a multidimensionalidade dos dados do *data warehouse*, a organização dos dados dessa base de dados em diferentes níveis de agregação e as características das consultas OLAP comumente realizadas pelos usuários de sistemas de suporte à decisão [CIFE02].

Nas próximas seções deste capítulo, são mostradas as propostas existentes atualmente de ambientes de *data warehouse* distribuído.

3.3 Arquitetura de Data Warehouse Distribuído de Inmon

A arquitetura de *data warehouse* distribuído definida por Inmon [INM02] é baseada nos conceitos de *data warehouse* local e de *data warehouse* global. A Figura 3.1 ilustra esta arquitetura, onde o *data warehouse* global situa-se localizado no *site* correspondente ao escritório central ou sede da empresa, enquanto os *data warehouses* locais estão localizados em regiões geográficas diferentes ou comunidades técnicas distintas.

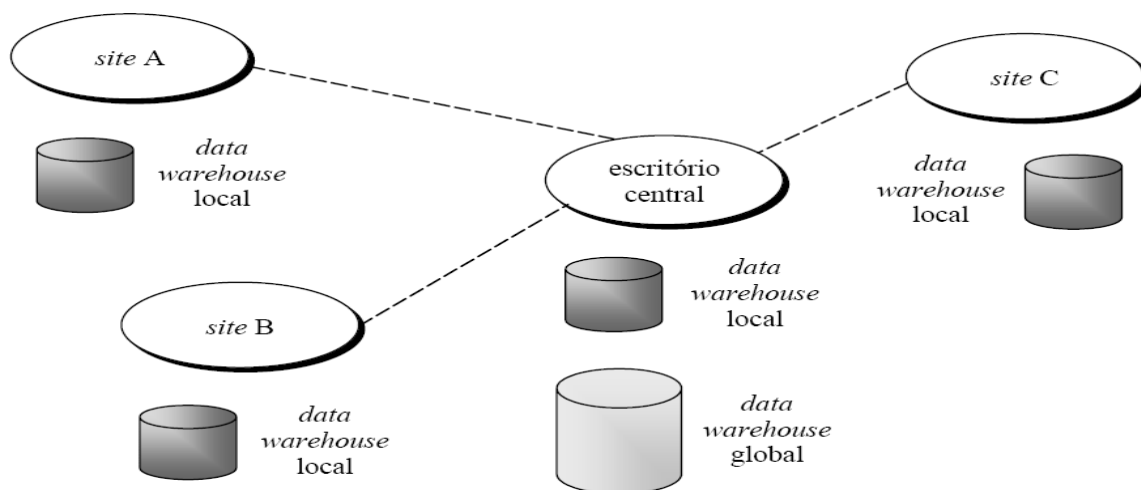


FIGURA 3.1 – ARQUITETURA BÁSICA DE DATA WAREHOUSE DISTRIBUÍDO DE INMON [CIFE02]

Os dados armazenados no *data warehouse* local são de interesse somente para o nível local, ou seja, cada *data warehouse* local tem como escopo dos seus dados os dados detalhados que refletem a integração das informações provenientes dos sistemas operacionais do *site* local ao qual ele serve. Apesar de ser inteiramente possível a existência de algum grau de compartilhamento entre os sistemas do ambiente operacional encontrados em cada um dos *sites*, qualquer interseção ou compartilhamento dos dados de um *data warehouse* local para outro é apenas uma coincidência [INM02].

Os dados armazenados no *data warehouse* global são de interesse para a empresa como um todo. Estes dados são integrados a partir das interseções naturais dos dados existentes nos *sites* que compõem o ambiente distribuído.

O relacionamento entre o *data warehouse* global e os *data warehouses* locais pode ser observado da seguinte forma. Os dados levemente agregados residem no nível global, enquanto que os dados detalhados residem nos níveis locais.

Como pode ser observado, os dados localizados no *data warehouse* global e nos *data warehouses* locais são mutuamente exclusivos: qualquer dado no *data warehouse* global não é encontrado nos *data warehouses* locais, e vice-versa. Em contrapartida, o projeto estrutural dos dados corporativos armazenados no *data warehouse* global pode sobrepor porções dos modelos de dados dos *data warehouses* locais.

Inmon [INM02] propõe uma variação desta arquitetura, onde consiste no pré-armazenamento dos dados a serem enviados ao *data warehouse* global por cada um dos *sites* locais. Assim, cada *site* que participa do ambiente armazena os dados globais correspondentes às informações locais em uma base de dados especial, chamada de área de armazenamento do *data warehouse* global, antes de enviá-los ao *data warehouse* global propriamente dito. Neste caso, a restrição de exclusividade mútua dos dados é observada tanto entre os dados localizados nos *data warehouses* locais e nas áreas de armazenamento do *data warehouse* global quanto entre os dados localizados nos *data warehouses* locais e no *data warehouse* global. Contudo, pode haver alguma redundância entre os dados armazenados no *data warehouse* global e nas áreas de armazenamento do *data warehouse* global em cada um dos *sites*, caso a política adotada pela empresa seja a não remoção dos dados destas áreas após o envio destes ao *data warehouse* global. A Figura 3.2 representa as áreas de armazenamento do *data warehouse* global em cada um dos *sites*.

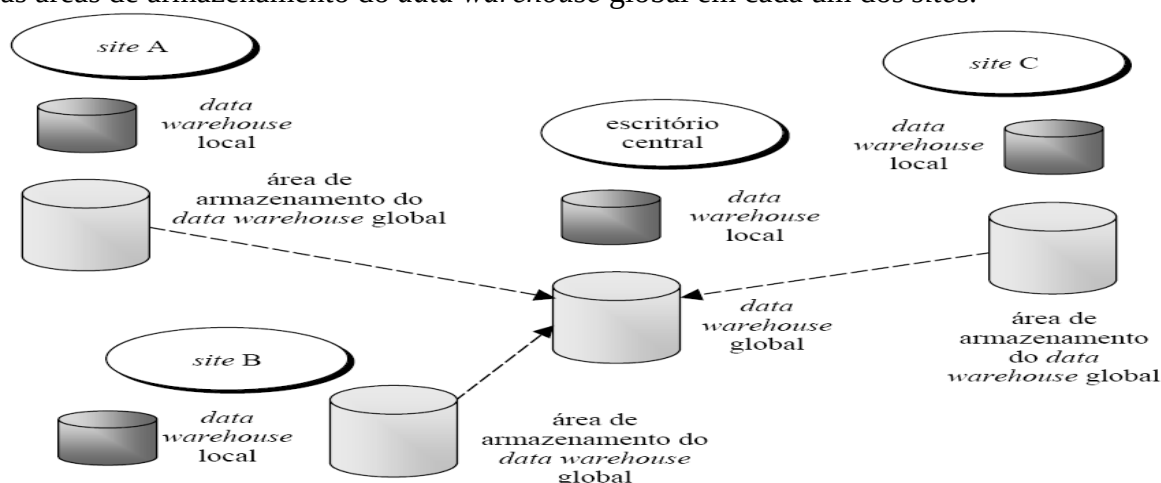


FIGURA 3.2 – VARIAÇÃO DA ARQUITETURA BÁSICA DE DATA WAREHOUSE DISTRIBUÍDO DE INMON [CIFE02]

Inmon [INM02] sugere que o desenvolvimento desta arquitetura deve ser feito primeiramente criando os *data warehouses* locais para cada entidade geográfica, para que depois, o *data warehouse* global seja criado, refletindo a integração dos negócios através das diferentes localizações.

3.4 Arquitetura de Data Warehousing Distribuído de Moeller

As arquiteturas de *data warehousing* distribuído definidas por Moeller [MOE01] são baseadas na junção de dois conceitos: integração através do elemento banco de dados e distribuição através do elemento rede. Assim, um *data warehouse* distribuído é definido por Moeller como uma coleção de dados compartilhados logicamente integrada, a qual é fisicamente distribuída através dos nós de uma rede de computadores.

Uma vez que o *data warehouse* distribuído consiste na integração lógica de diversos bancos de dados locais, ele não existe fisicamente nas arquiteturas de Moeller. Mais especificamente, o *data warehouse* distribuído é apenas um conceito virtual. Em particular, os termos local e global são utilizados para realizar a distinção, respectivamente, entre os aspectos relacionados a um único *site* e os aspectos relacionados ao ambiente de *data warehousing* como um todo. Por exemplo, um *data warehouse* local refere-se a um banco de dados pré-existente que reside em um *site* específico da rede, ou seja, refere-se a um *data mart*.

Há três diferentes tipos de arquitetura de *data warehousing* distribuído apresentadas por Moeller [MOE01]: arquitetura de *data warehousing* distribuído homogêneo, heterogêneo e com um SGBD distribuído único. Esses tipos serão abordados nas próximas subseções.

É importante salientar que Moeller associa os seus três tipos de arquitetura de *data warehousing* distribuído à abordagem de desenvolvimento, na qual uma corporação já gerencia vários *data marts* independentes e deseja possibilitar, como uma atividade subsequente, o acesso global dos usuários de SSD a estes *data marts* através de um *data warehouse* global virtual. Ou seja, os dados são mantidos nas fontes de dados e as consultas são decompostas em tempo real e submetidas às diversas fontes, onde o resultado é integrado e mostrado para o usuário que efetuou a consulta. Isto é obtido através do desenvolvimento de um esquema global da empresa como um todo, que representa a integração dos esquemas locais dos *data marts* existentes, além da interconexão destes *data marts* através da rede.

3.4.1 Arquitetura de Data Warehousing Distribuído Homogêneo

A Figura 3.3 mostra a arquitetura de *data warehousing* distribuído homogêneo proposta por Moeller [MOE01], com os seus dois principais componentes: o *data warehouse* distribuído e a ferramenta de banco de dados distribuídos.

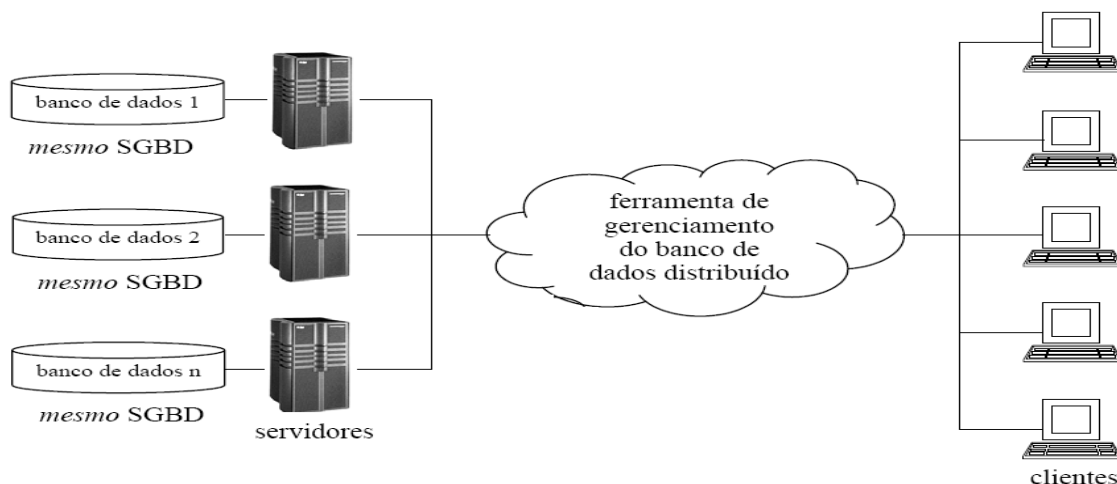


FIGURA 3.3 – ARQUITETURA DE DATA WAREHOUSING DISTRIBUÍDO HOMOGÊNEO DE MOELLER [CIFE02]

Como visto anteriormente, cada *site* nesta arquitetura possui o seu próprio banco de dados autônomo e pode representar um *data mart* independente. A arquitetura homogênea é caracterizada por apresentar em todos os *sites* o mesmo SGBD. São nestes SGBD que se armazenam os *data marts* a serem distribuídos.

A ferramenta de gerenciamento do banco de dados distribuído, por sua vez, é responsável por integrar os diversos bancos de dados locais, oferecendo uma visão lógica do *data warehouse* corporativo, além de gerenciar as consultas dos usuários de SSD aos bancos de dados fora de suas redes locais. Essa ferramenta é baseada em dois elementos centrais relacionados à manipulação dos dados distribuídos: esquema de fragmentação e esquema de alocação. O esquema de fragmentação descreve como os relacionamentos globais são divididos entre os bancos de dados locais. Já o esquema de alocação especifica a localização de cada um dos fragmentos, possibilitando a execução de consultas através dos diversos bancos de dados locais. Este último esquema também dá suporte à possibilidade de replicação dos dados na arquitetura.

3.4.2 Arquitetura de Data Warehousing Distribuído Heterogêneo

A arquitetura de *data warehousing* distribuído heterogêneo proposta por Moeller [MOE01] é baseada nos mesmos componentes principais que a arquitetura de *data warehousing* distribuído homogêneo: o *data warehouse* distribuído e uma ferramenta de gerenciamento do banco de dados distribuído. No entanto, na arquitetura de *data warehousing* distribuído heterogêneo, estes componentes possuem características e funcionalidades particulares relacionadas à heterogeneidade dos dados, aumentando, com isso, a complexidade destes componentes. A Figura 3.4 ilustra esta arquitetura.

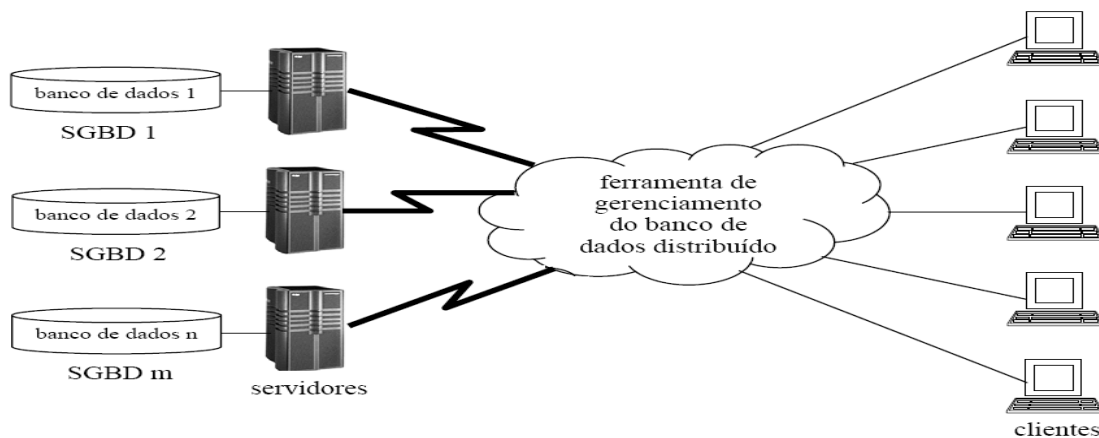


FIGURA 3.4 – ARQUITETURA DE DATA WAREHOUSING DISTRIBUÍDO HETEROGÊNEO DE MOELLER [CIFE02]

Cada *site* nesta arquitetura possui o seu próprio banco de dados autônomo e pode representar um *data mart* independente. A arquitetura heterogênea possibilita que diferentes SGBD sejam utilizados nos *sites* da arquitetura, para armazenar os bancos de dados a serem distribuídos. É de responsabilidade da ferramenta de gerenciamento do banco de dados distribuído tratar e oferecer os serviços adicionais voltados ao tratamento da heterogeneidade.

Além desses serviços adicionais, as demais funcionalidades da ferramenta de gerenciamento do banco de dados distribuído na arquitetura de *data warehousing* distribuído heterogêneo são as mesmas funcionalidades oferecidas por essa ferramenta na arquitetura homogênea:

- Conectar os diversos bancos de dados independentes através de uma rede de computadores, oferecendo uma visão lógica integrada dos dados corporativos;
- Atender às consultas dos usuários de SSD que requisitam dados através dos *sites* da arquitetura; e
- Proporcionar os esquemas de fragmentação e de alocação.

É essencial a presença de um modelo de dados global integrado para o bom funcionamento da ferramenta de gerenciamento do banco de dados distribuído.

3.4.3 Arquitetura de Data Warehousing Distribuído com SGBD Distribuído Único

A Figura 3.5 mostra a arquitetura de *data warehousing* distribuído proposta por Moeller [MOE01]. Diferentemente do que foi visto nas arquiteturas de *data warehousing* distribuído homogêneo e heterogêneo, na arquitetura com SGBD distribuído único não existem banco de dados locais autônomos, ou seja, esta arquitetura não oferece suporte a *data marts* independentes.

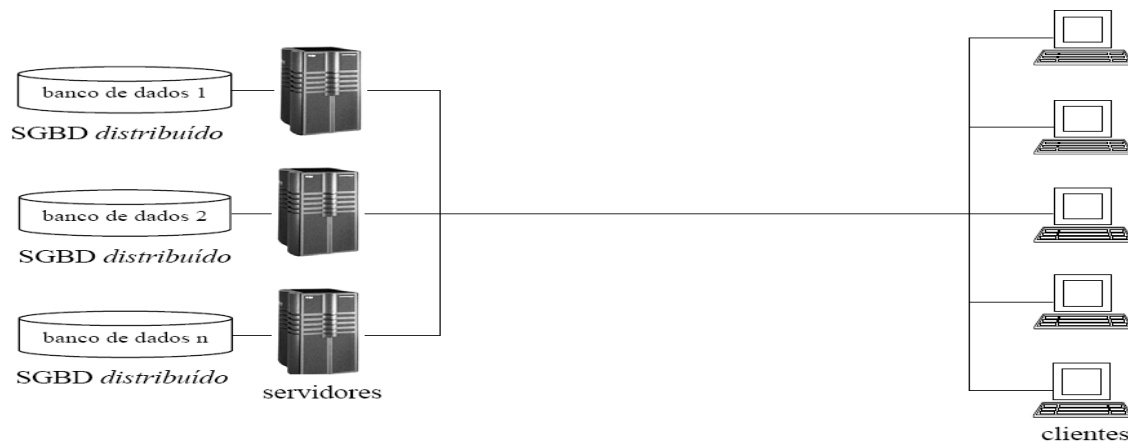


FIGURA 3.5 – ARQUITETURA DE DATA WAREHOUSING DISTRIBUÍDO COM SGBD DISTRIBUÍDO ÚNICO DE MOELLER [CIFE02]

Nesta arquitetura, os dados do *data warehouse* corporativo podem estar armazenados em diferentes *sites*, podendo ser distribuídos (fragmentados e/ou replicados) nestes *sites* à medida que o volume do *data warehouse* aumenta ou à medida que o número de usuários cresce. O acesso a estes dados é feito através do SGBD distribuído, que desempenha papel similar ao exercido pela ferramenta de gerenciamento do banco de dados distribuído nas arquiteturas de *data warehousing* distribuído homogêneo e heterogêneo, fazendo-se desnecessária a presença desta ferramenta nesta arquitetura.

Enquanto nas arquiteturas homogênea e heterogênea cada banco de dados local possui o seu próprio modelo de dados individual, na arquitetura com SGBD distribuído único não existem modelos de dados locais. Tal restrição está relacionada ao fato de que as porções do *data warehouse* corporativo armazenadas nos diversos *sites* dessa última arquitetura não podem ser consideradas bancos de dados locais autônomos. Apesar disto, é indispensável a definição de um modelo de dados corporativo na arquitetura com SGBD distribuído único [MOE01].

3.4.4 Abordagem para o Problema de Localização dos Dados

Desde que diversos *data marts* pré-existentes sejam integrados logicamente entre si através de uma rede a fim de criar um *data warehouse* corporativo, Moeller [MOE01] afirma que pode ser interessante realizar uma redistribuição posterior dos dados destes *data marts*, considerando-se requisitos globais de fragmentação, de replicação e de alocação, visando-se um aumento de desempenho do sistema como um todo.

Moeller apresenta uma abordagem prática para o problema de localização dos dados, que é dividida em duas partes. A primeira parte desta abordagem discute aspectos relacionados à localização dos dados de forma a maximizar o desempenho do sistema no atendimento às consultas dos usuários de SSD. A segunda parte da abordagem proposta identifica técnicas que visam minimizar o tempo gasto no carregamento dos dados.

Segundo Ciferri [CIFE02], os doze passos sequenciais da primeira parte da abordagem proposta por Moeller são:

- Passo 1 - definir os objetivos particulares e as características físicas do *data warehouse* distribuído;
- Passo 2 - identificar as características dos SGBD utilizados para implementar os bancos de dados locais de forma que estas características também sejam seguidas no projeto do *data warehouse* distribuído. Este passo aplica-se somente às arquiteturas de *data warehousing* distribuído homogêneo e heterogêneo;
- Passo 3 - ajustar (maximizar) o desempenho individual dos bancos de dados locais. A importância desta tarefa está relacionada ao fato de que, segundo Moeller, a maioria das consultas dos usuários de SSD é processada por *data marts* particulares em um *data warehouse* distribuído;
- Passo 4 - realizar a manutenção periódica dos bancos de dados locais;
- Passo 5 - otimizar o código gerado por consultas que acessam dados localizados em diferentes *sites* da arquitetura de *data warehousing* distribuído;
- Passo 6 - aplicar técnicas de otimização padrão para aumentar o desempenho no processamento de consultas complexas. Por exemplo, pode-se optar pela incorporação de um índice ou pelo particionamento de uma tabela grande;
- Passo 7 - identificar o relacionamento existente entre os dados, de forma que os dados fortemente vinculados (ou seja, os dados frequentemente requisitados conjuntamente) sejam armazenados em um mesmo *site*;
- Passo 8 - identificar os dados predominantemente estáticos, ou seja, os dados não-voláteis ou os dados que são alterados esporadicamente. Esses dados devem ser localizados no *site* no qual são mais requisitados. Em especial, dados estáticos são passíveis de replicação;
- Passo 9 - efetuar o particionamento vertical dos dados do *data warehouse* distribuído. Isto pode ser realizado através da fragmentação vertical de uma tabela existente em duas ou mais tabelas, e da posterior alocação dessas novas tabelas aos *data marts* adequados. Isto também pode ser realizado através da unificação de diferentes tabelas que compartilham chaves primárias, e da posterior alocação da tabela resultante a um *site* específico. Nesse caso, os dados duplicados são removidos;
- Passo 10 - efetuar o particionamento horizontal dos dados do *data warehouse* distribuído. Isto pode ser realizado através da fragmentação horizontal de uma tabela grande em duas ou mais tabelas, e da posterior alocação dessas tabelas aos *data marts* adequados;
- Passo 11 - replicar tabelas ou fragmentos de tabelas, de forma a minimizar o tráfego de dados pela rede no suporte a consultas OLAP. Isto deve ser realizado de forma a balancear o ganho de desempenho no processamento de consultas OLAP com o aumento dos custos de manutenção; e
- Passo 12 - redistribuir os dados do *data warehouse* distribuído, os processos e até mesmo os usuários de determinados *data marts*, visando reduzir gargalos existentes.

A segunda parte da abordagem para o problema de localização dos dados proposta por Moeller visa otimizar o tempo de carregamento dos dados. Nela, discutem-se os aspectos práticos relacionados: às ferramentas e/ou aos algoritmos empregados no carregamento dos dados, ao pré-processamento dos dados, à possibilidade de carregamento dos dados em paralelo e ao balanceamento de processamento na fase de carregamento dos dados entre os *data marts* da arquitetura de *data warehousing* distribuído, dentre outros.

3.5 Arquitetura de Data Warehouse Distribuído Hierárquico de Zhou et al.

Zhou et al. [ZZTH00] antes de descreverem sua arquitetura de *data warehouse* distribuído hierárquico, primeiramente descrevem um estudo de caso bem simples de *data warehouse* modelado com o esquema estrela, ilustrado na Figura 3.6, contendo três tabelas de dimensões (Área, Tempo e Indústria) e uma tabela de fatos, apresentando uma medida representada pelo atributo GNP, que mede a performance da economia.



FIGURA 3.6 – ESQUEMA ESTRELA DA INFORMAÇÃO ECONÔMICA DO DATA WAREHOUSE

Zhou et al. [ZZTH00] destacam as desvantagens de se utilizar tanto o *data warehouse* centralizado (Figura 3.7), quanto o *data warehouse* distribuído ou federado proposto por Wells et al. [WDCP96] (Figura 3.8). As desvantagens encontradas na construção deste enorme *data warehouse* em um ambiente centralizado são: a dificuldade de carga e manutenção desta enorme quantidade de dados e a grande quantidade de usuários que farão consultas aos servidores OLAP.



FIGURA 3.7 – DW CENTRALIZADO (BASEADO EM [AJLW98])

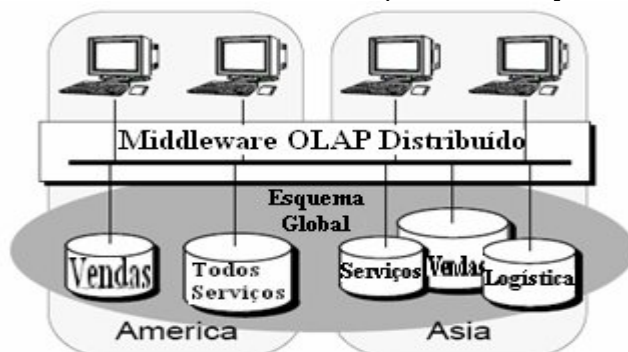


FIGURA 3.8 – DW DISTRIBUÍDO OU FEDERADO (BASEADO EM [AJLW98])

Já o ambiente distribuído de Wells et. al [WDCP96] oferece uma oportunidade de construção de um largo *data warehouse* corporativo integrado, utilizando-se de um *middleware* OLAP distribuído para realizar as consultas OLAP e integrá-las para serem retornadas ao usuário. O esquema global utilizado nesta arquitetura serve para ter uma visão integrada dos esquemas dos *data marts* existentes.

A desvantagem desta arquitetura, segundo Zhou et al, é que os *data marts* estão em um único nível, impossibilitando dar atenção a níveis de usuários de alto e baixo nível ao mesmo tempo. O Quadro 3.1 compara os três tipos de arquiteturas de *data warehouse*.

QUADRO 3.1 – COMPARAÇÕES ENTRE DW CENTRALIZADO, DW DISTRIBUÍDO DE WELLS ET AL. E HDDW (BASEADO EM [ZZTH00])

Características	DW Centralizado	DW Distribuído	HDDW
Distribuição dos Dados	Baixo	Moderado	Alto
Execução da Consulta	Fácil	Complexo	Moderado
Custo de Comunicação	Baixo	Alto	Moderado
Manutenção	Difícil	Moderado	Fácil

E com isso eles propõem como uma melhor alternativa para este estudo de caso, a sua arquitetura de *data warehouse* distribuído hierárquico (HDDW). Esta arquitetura tem duas características “especiais”, são elas [ZZTH00]:

1. Cada departamento em todos os níveis deve ter o seu próprio *data warehouse* ou *data mart*; e
2. Cada departamento cuida principalmente do dado econômico da área em que se localiza, isto é, a tomada de decisão do departamento com nível estado se preocupa basicamente com o dado econômico agregado do estado e os dados detalhados das cidades dos municípios que fazem parte deste estado. E assim por diante com os outros níveis da arquitetura.

A construção da arquitetura HDDW é feita gradualmente, onde se utiliza a abordagem *bottom-up* de desenvolvimento do *data warehouse*, construindo *data marts* no nível mais baixo da arquitetura e subindo gradativamente o nível de desenvolvimento. A Figura 3.9 ilustra esta arquitetura dividida em níveis de hierarquias de relacionamento entre os atributos da dimensão Área (Figura 3.6), onde o desenvolvimento começa no nível de bairro e sobe até o nível nação que corresponde à integração dos *data marts* existentes no nível de estado.

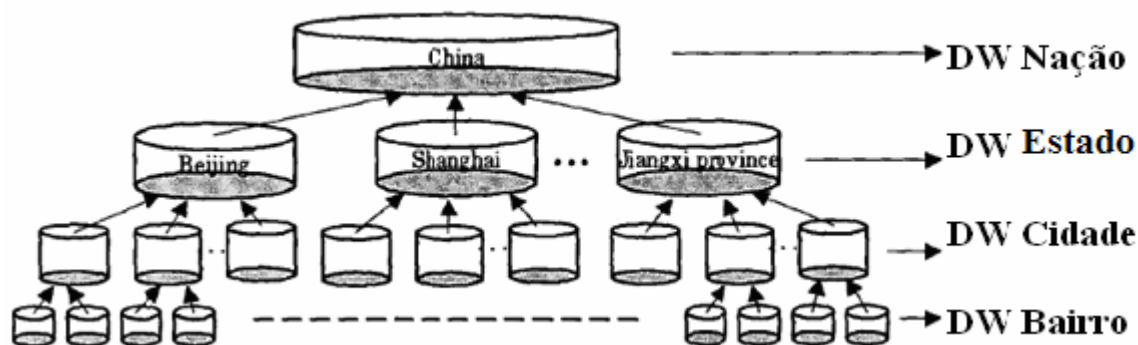


FIGURA 3.9 – DATA WAREHOUSE DISTRIBUÍDO HIERÁRQUICO (BASEADO EM [ZZTH00])

Como visto anteriormente, o desenvolvimento dos *data marts* na arquitetura HDDW é construída nível por nível. Com isso, a implementação e a manutenção fica realmente fácil, devido aos *data marts* serem praticamente independentes entre si, portanto, a manutenção no Quadro 3.1 pode ser considerada fácil nesta arquitetura. A manutenção acontece quando um dado é alterado em um nível e esta alteração se reflete a todos os *data marts* que contenham referência para este dado de níveis acima até chegar ao topo [ZZTH00].

O estudo de caso mostra que se pode distribuir os dados de acordo com níveis de hierarquias de relacionamento entre os atributos de uma dimensão. Como exemplo, neste estudo de caso podia-se distribuir os dados usando os atributos da dimensão Tempo: ano, estação e mês para formar uma hierarquia de dados.

3.5.1 HDDW orientado a consultas OLAP

A Figura 3.10 ilustra a arquitetura do HDDW orientada a consultas OLAP. Pode-se observar nesta figura que os usuários finais utilizam uma ferramenta OLAP comum e um servidor OLAP especializado, que será mostrada na Figura 3.11. Todos estes servidores OLAP conectam-se entre si através da rede e operam colaborativamente para responder as consultas dos usuários do SSD [ZZTH00].

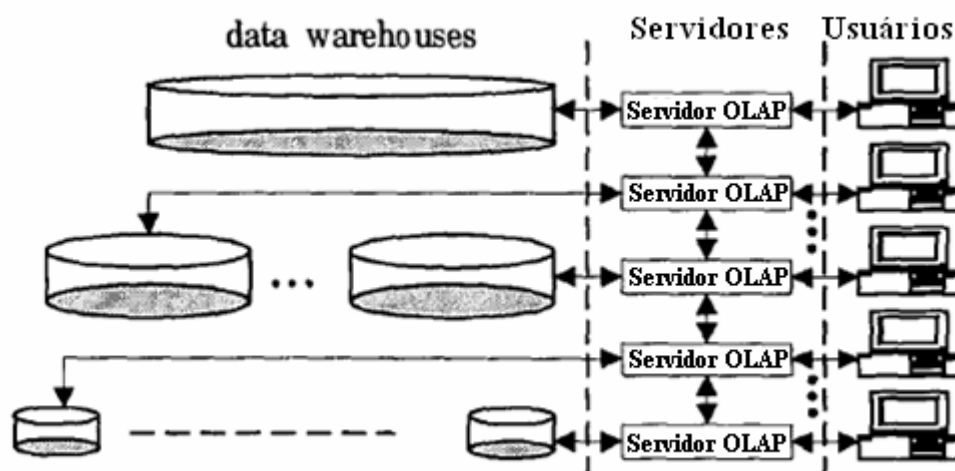


FIGURA 3.10 - HDDW ORIENTADO A CONSULTAS OLAP (BASEADO EM [ZZTH00])

A Figura 3.11 mostra o mecanismo de processamento das consultas OLAP na arquitetura HDDW. Este mecanismo é composto por três camadas: cliente, servidor OLAP local (middleware) e servidor de banco de dados. A camada cliente oferece para o usuário final ferramentas com facilidades para gerar relatórios. A camada do *middleware* é a parte principal da arquitetura, que esconde os detalhes da distribuição dos dados e tem como principal objetivo a geração otimizada da consulta distribuída para responder com transparência a consulta feita pelo usuário. A camada do servidor de banco de dados é quem possibilita o acesso aos dados existentes nos *data marts*.

O cliente acessa o sistema por meio do servidor OLAP. A ferramenta usada pelo usuário provê interatividade com o ambiente do servidor OLAP local. As consultas

requisitadas pelo usuário são transferidas para o servidor local OLAP em uma linguagem SQL formal.

Quando uma consulta é recebida pelo servidor OLAP local, ela é analisada sintaticamente e depois é verificado no esquema global quais *data marts* devem ser consultados para responder a consulta, com isso a consulta é decomposta em sub-consultas remotas e local. Finalmente, os resultados das sub-consultas são integrados e submetidos ao usuário. A máquina de agregação é responsável pelo cálculo agregado e o gerenciamento, que é um fator chave para o desempenho em servidores OLAP [ZZTH00].

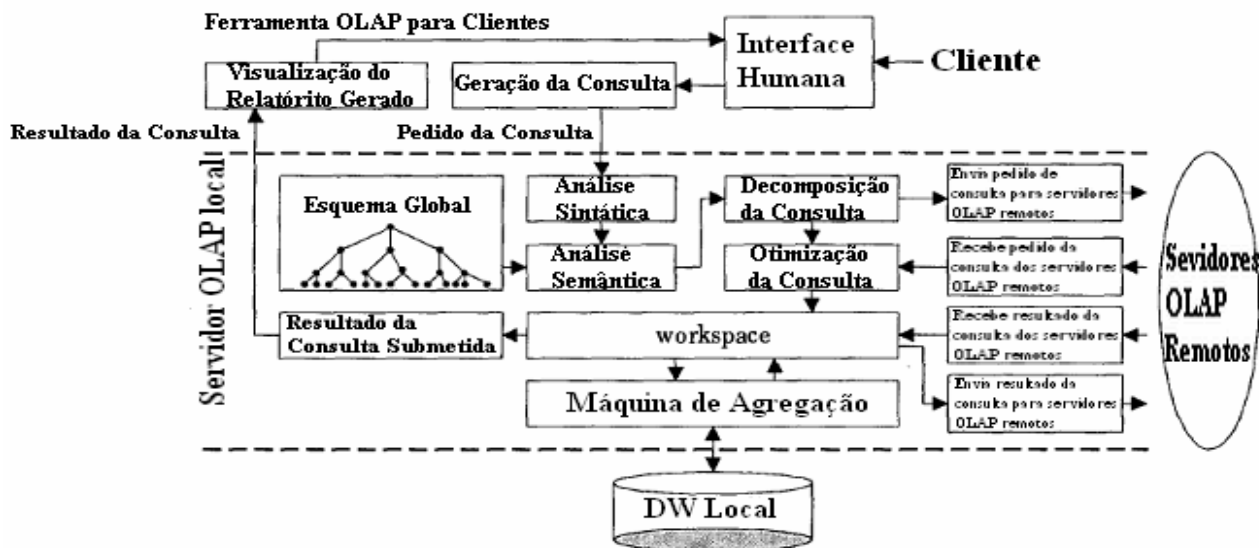


FIGURA 3.11 – MECANISMO OLAP DO SISTEMA HDDW (BASEADO EM [ZZTH00])

O gerenciamento dos usuários do sistema é feito através do agrupamento dos usuários em determinados níveis de hierarquia. Caso o nível do usuário seja o mais alto da hierarquia, ele terá acesso a todos os dados do sistema; já, se for o mais baixo, só terá acesso aos dados daquele nível.

3.6 Arquitetura de Data Warehouse Distribuído de Ciferri (O Sistema WebD²W)

O sistema WebD²W (Web Distributed Data Warehousing) consiste em um ambiente de *data warehousing* distribuído cliente-servidor que visa não somente à distribuição dos dados do *data warehouse*, mas também ao acesso distribuído a esses dados usando a tecnologia Web como infra-estrutura [CCSF02].

Este sistema utiliza-se da distribuição dos dados do *data warehouse*, visando alcançar diversas vantagens sobre a arquitetura de um *data warehouse* centralizado, tais como: aumento da disponibilidade dos dados, melhoria na confiabilidade, suporte a um grande número de usuários, melhoria no desempenho das consultas OLAP e uma expansão mais fácil das unidades de distribuição. Além destas vantagens, o sistema garante ainda: a

transparência de fragmentação, de replicação e de localização na manipulação dos dados, tornando as consultas OLAP transparentes para os usuários dos SSD.

3.6.1 Arquitetura

A arquitetura de um ambiente de *data warehousing* centralizado, mostrada na Figura 2.6 deste trabalho, é estendida para a arquitetura do sistema WebD²W, porém, a inclusão dos componentes de distribuição, o componente de consulta do ambiente distribuído e o componente de manutenção do ambiente distribuído fazem com que os outros componentes do sistema apresentem alterações em suas estruturas internas, aumentando a complexidade no desenvolvimento dos mesmos.

A arquitetura do sistema WebD²W é mostrada na Figura 3.12, onde são identificados todos os componentes, os inter-relacionamentos entre eles e as classes de usuários existentes no sistema. As funcionalidades existentes em cada componente e as responsabilidades de cada classe de usuário serão descritas nas próximas seções e subseções desse capítulo.

As informações estruturais e semânticas das fontes de dados e outras informações que possam ser importantes para o ambiente de *data warehousing* são armazenadas no repositório de metadados, sendo manipuladas pelo gerenciador de metadados [CIFE02].

O componente de gerenciamento e administração do ambiente tem funcionalidades voltadas para o monitoramento do ambiente de *data warehousing* distribuído. Tanto o administrador do *data warehouse* distribuído quanto o administrador do *data warehouse* global interagem com este componente.

A abordagem utilizada para o desenvolvimento do sistema WebD²W foi a *top-down*, onde o *data warehouse* global foi construído primeiro, para que, posteriormente, as unidades de distribuição fossem geradas. Estas unidades de distribuição podem ser vistas como *data warehouses* ou *data marts*, dependendo dos critérios utilizados para a distribuição dos dados.

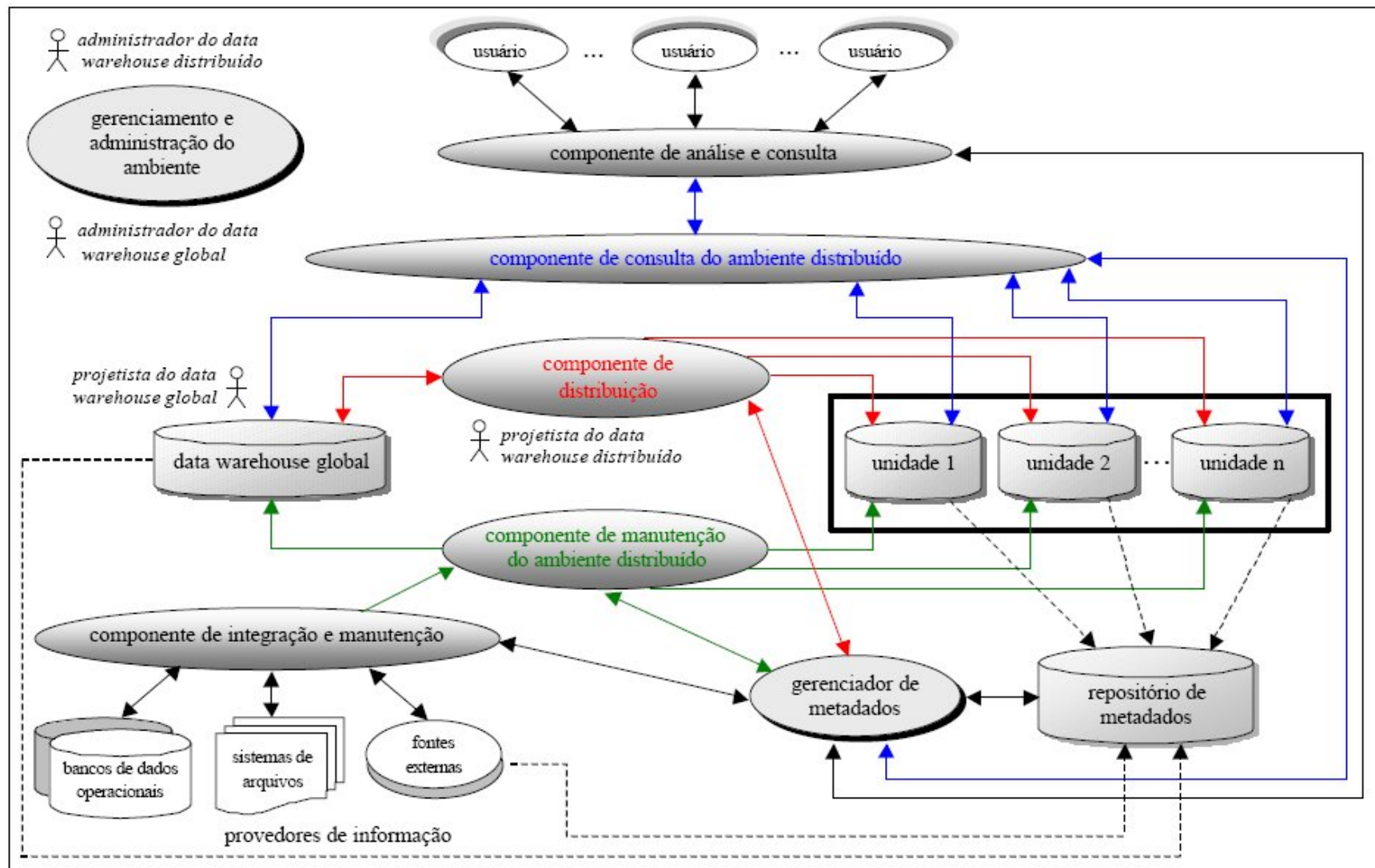


FIGURA 3.12 – ARQUITETURA BÁSICA DO SISTEMA WebD²W [CIFE02]

3.6.2 A importância do Data Warehouse Global

O sistema WebD²W define o *data warehouse* global como sendo o *data warehouse* que contém todos os dados que serão distribuídos para os diversos *sites*. Os dados serão extraídos dos provedores de informação (banco de dados operacionais, sistemas de arquivos e fontes externas) e depois traduzidos, limpados e integrados pelo componente de integração e manutenção, para que se possam carregar os dados no *data warehouse* global.

A importância do *data warehouse* global é observada pelo fato de que, como ele contém todos e os mais atualizados dados do sistema é a partir dele que são gerados fragmentos que serão alocados nas várias unidades de distribuição do sistema, evitando a necessidade da comunicação direta entre unidades de distribuição e provedores de informação. O oposto resultaria em um custo adicional elevado ao sistema, devido à fase de carregamento dos dados ser um processo de atividades extremamente complexas e lentas [NASC05]. Além disso, as consultas OLAP poderiam ocasionar um grande volume de dados trafegando pela rede, sobrecarregando-a e, possivelmente, piorando a performance destas consultas.

Nas próximas subseções, são descritos brevemente os cinco componentes presentes na arquitetura do sistema WebD²W, que são: componente de integração e manutenção, componente de distribuição, componente de análise e consulta, componente de consulta do ambiente distribuído e o componente de manutenção do ambiente distribuído.

3.6.3 Componente de Integração e Manutenção

É responsável pela integração dos dados existentes nas fontes de dados, após a extração tradução, limpeza, filtragem e integração dos dados, além de atualização periódica e expiração dos dados.

Como pode ser observado, este componente desempenha as mesmas funções dos componentes vistos na seção 2.6.1. Porém, no sistema WebD²W, o componente de integração e manutenção é estendido de forma a possibilitar a propagação das alterações a serem realizadas nos dados do *data warehouse* global ao componente de manutenção do ambiente distribuído.

3.6.4 Componente de Manutenção do ambiente distribuído

É responsável por manter a consistência dos dados no *data warehouse* global e nas unidades de distribuição. Segundo Ciferri [CIFE02], este componente pode desempenhar duas funcionalidades diretamente relacionadas:

- A manutenção da consistência intra-site – A consistência intra-site resulta da habilidade de se manter os dados detalhados e agregados armazenados em um *site* particular consistentes entre si; e
- A manutenção da consistência entresites - A consistência entresites resulta da habilidade de se manter os dados fragmentados e/ou replicados através dos diversos *sites* do ambiente de *data warehousing* consistentes entre si.

3.6.5 Componente de Distribuição

Tem como principal objetivo aumentar a disponibilidade dos dados do *data warehouse* global, alocando os dados replicados e/ou fragmentados através dos diversos *data warehouses* distribuídos no sistema. Este componente também é responsável pelo aumento da confiabilidade e o do suporte a um maior número de usuários no sistema. O componente de distribuição é composto por quatro módulos: requisitos, fragmentação, alocação e carga que serão descritos nos subtópicos abaixo.

3.6.5.1 Módulo Requisitos

Tem como responsabilidade determinar um conjunto de critérios que deve ser utilizado como base pelo projetista do *data warehouse* distribuído para definir limitações a serem aplicadas aos processos de fragmentação, replicação e alocação dos dados do *data warehouse* global. São considerados requisitos para o processo de fragmentação e alocação a obtenção de informações a respeito dos sistemas computacionais, da rede de comunicação, da carga de trabalho e do banco de dados.

As informações dos sistemas computacionais requerem um conhecimento dos *sites* que serão utilizados pelo ambiente distribuído para alocação das unidades de distribuição: características como nível de segurança oferecido, capacidade de processamento de hardware e armazenamento de dados, software utilizado, entre outros.

Devem ser obtidas também informações da rede de comunicação, como largura de banda da rede, para melhor alocar os dados mais acessados em *sites* com grande largura de banda e segurança da rede. Já as informações a respeito da carga de trabalho consistem em saber qual porção de dados é acessada mais frequentemente para que uma análise sobre eles seja traçada e possa resultar em uma fragmentação e alocação adequadas, e que realmente aumente a disponibilidade dos dados consideravelmente. E por fim, informações relativas ao banco de dados, como suas estruturas e características, de maneira geral [COSTA05].

3.6.5.2 Módulo Fragmentação

Este módulo tem como objetivo o particionamento dos dados do *data warehouse* global em diversos fragmentos, para que, posteriormente, estes dados sejam alocados nas diversas unidades de distribuição existentes. O particionamento dos dados pode ser feito de três diferentes tipos de fragmentação: a horizontal, a vertical e a mista.

A fragmentação horizontal divide cada fragmento em um subconjunto das tuplas da relação completa, onde cada tupla de uma relação precisa ser armazenada em pelo menos uma unidade de distribuição. Na fragmentação vertical, as relações são decompostas em relação aos atributos, onde cada fragmento é uma projeção da relação completa e cada fragmento tem, necessariamente, como atributo, a chave primária ou algum atributo que seja chave candidata da relação completa. A fragmentação mista é a combinação da fragmentação horizontal e vertical, mas sempre uma por vez, ou seja, ou se fragmenta verticalmente para depois fragmentar horizontalmente, ou o contrário.

Este módulo faz uso tanto do algoritmo de fragmentação horizontal proposto por Ciferri [CIFE02], quanto pelos algoritmos de fragmentação vertical propostos por

Nascimento [NASC05] e Costa [COSTA05], que gerará entradas para o módulo de alocação.

Segundo Ciferri [CIFE02], “sendo considerada uma implementação relacional das visões materializadas no *data warehouse* através do esquema estrela, pode-se afirmar que as técnicas de fragmentação horizontal, vertical e mista devem focar principalmente a fragmentação da tabela de fatos, sendo que a fragmentação das tabelas de dimensão é inerente à fragmentação da tabela de fatos”.

3.6.5.3 Módulo Alocação

O módulo alocação tem como objetivo alocar os fragmentos resultantes do módulo fragmentação nas unidades de distribuição. Também é responsabilidade deste módulo escolher quais destes fragmentos devem ser replicados e em qual *sites* estas réplicas serão alocadas, aumentando, com isso, a disponibilidade e eficiência das consultas OLAP dos usuários de SSD. Porém, a replicação pode reduzir a velocidade das operações de atualização, uma vez que, uma atualização deve ser executada em todas as cópias do banco de dados, mantendo a consistência dos seus dados.

Em adição aos custos de manutenção, o problema de alocação também deve considerar os custos de armazenamento dos fragmentos nos *sites*, os custos associados ao processamento das consultas e os custos de transmissão. É importante destacar que o problema de alocação consiste em um problema de otimização muito complexo [ERNS04].

3.6.5.4 Módulo Carga

O módulo carga tem como principal objetivo o carregamento inicial dos dados nos diversos *sites*, utilizando-se do esquema de fragmentação gerado pelo módulo fragmentação e do esquema de alocação gerado pelo módulo alocação para este propósito. A transferência dos dados do *data warehouse* global para as unidades de distribuição pode ser realizada através da Web ou de forma indireta, quando o volume de dados inviabiliza o uso da rede [CIFE02].

O módulo carga desempenha também outras funcionalidades complementares ao carregamento inicial dos dados, são elas:

- Armazenamento do projeto do ambiente distribuído no repositório de metadados;
- Criação e carga do repositório de metadados nos *sites* das unidades de distribuição; e
- Responsável por inicializar os demais componentes do sistema

Este módulo pode ser ativado tanto pelo carregamento inicial dos dados nos *sites* quanto pela criação de novos *sites* no sistema WebD²W. Neste último caso, as informações relativas aos esquemas destas novas unidades devem ser replicadas em todos os repositórios de metadados presentes.

3.6.6 Componente de Consulta do Ambiente Distribuído

O componente de consulta do ambiente distribuído tem como principal objetivo o aumento da disponibilidade do acesso aos dados relativos à consulta. Este componente pode ser representado de acordo com duas perspectivas:

- Funcionalidade de Acesso Local – Explora a proximidade do dado buscado na consulta em relação à localidade do usuário, otimizando, com isso, a consulta. Nesta funcionalidade só são consideradas as consultas submetidas ao sistema WebD²W via Intranet, ou seja, o usuário de SSD que fez a consulta tem que estar em um dos *sites* das unidades de distribuição ou no *site* que contém o *data warehouse* global; e
- Funcionalidade de Acesso Global – Explora também a proximidade do dado buscado na consulta em relação à localidade do usuário, porém, nesta funcionalidade só são consideradas as consultas submetidas ao sistema WebD²W oriundas da Internet. Nesse último caso, o usuário de SSD encontra-se localizado em um *site* que não armazena dados do *data warehouse* distribuído.

Através da replicação e distribuição destas funcionalidades nos diversos *sites* do ambiente distribuído é possível realizar acessos paralelos ao sistema WebD²W, além de se eliminar gargalos que porventura poderiam existir, caso essas funcionalidades fossem centralizadas em apenas um *site*.

Além disso, o acesso a esses dados deve ser de forma transparente, ou seja, o sistema deve garantir a transparência de fragmentação, replicação e alocação. Os usuários executam consultas ao sistema como se esse ambiente tivesse apenas um *data warehouse* centralizado [NASC05].

Por fim, este componente é responsável por redirecionar a consulta ao *site* mais apropriado em respondê-la, ou simplesmente gerenciar o processamento distribuído da mesma.

3.6.7 Arquitetura do Sistema WebD²W na Web

A arquitetura proposta pelo Sistema WebD²W consiste em uma arquitetura de genérica três camadas, a qual representa a solução adotada para o acesso aos dados do *data warehouse* distribuído no sistema.

A arquitetura de três camadas é composta por camada de apresentação, camada lógica e camada de gerenciamento de dados. Cada uma destas camadas é detalhada abaixo.

A camada de apresentação é responsável pela definição da lógica da interface gráfica, onde se determina quais dados e de que forma estes dados serão visualizados. Além disto, os valores digitados pelo usuário na interface gráfica devem ser validados e verificados. A camada lógica é onde se implementa a lógica do negócio, com isso é possível, através desta camada, acessar diferentes SGBD. A camada de gerenciamento de dados é onde residem os SGBD, que é responsável pelo armazenamento, gerenciamento e recuperação dos dados da aplicação.

4. Análise Comparativa de Arquiteturas de Distribuição de Data Warehouse

Neste capítulo é apresentada uma análise comparativa entre as arquiteturas de distribuição de *data warehouse*, descritas no capítulo 3, destacando-se vantagens e desvantagens de cada uma destas arquiteturas. Em seguida, baseando-se no resultado da análise são sugeridas algumas boas práticas para o desenvolvimento dessas arquiteturas.

Nesta análise comparativa têm-se como base uma extensão dos critérios definidos por Ciferri [CIFE02]:

- Aspectos para fragmentação - Se a arquitetura identifica ou discute questões importantes que poderiam ser utilizadas como base para a fragmentação dos dados;
- Aspectos para alocação e/ou replicação - Se a arquitetura identifica ou discute questões importantes que poderiam ser utilizadas como base para alocação e/ou replicação dos dados;
- Metodologia e/ou algoritmos para fragmentação - Se a arquitetura apresenta metodologias e/ou algoritmos para fragmentação dos dados;
- Metodologia e/ou algoritmos para alocação e/ou replicação - Se a arquitetura apresenta metodologias e/ou algoritmos para alocação e/ou replicação dos dados;
- Redirecionamento de consultas processadas de forma centralizada - Se a arquitetura apresenta o redirecionamento de consultas ao *site* mais apropriado em respondê-la de forma centralizada;
- Gerenciamento de consultas processadas de forma distribuída - Se a arquitetura apresenta o gerenciamento de consultas decompostas em diversas subconsultas, de forma que cada uma dessas subconsultas acesse diferentes partes do *data warehouse* distribuído;
- Controle local aos *sites* - Se a arquitetura proporciona controle de acesso local a cada *site* que participa do ambiente distribuído;
- Controle global aos *sites* - Se a arquitetura proporciona controle de acesso global ao ambiente distribuído como um todo;
- Manutenção da consistência - Se a arquitetura apresenta metodologias e/ou algoritmos para a manutenção da consistência dos dados distribuídos;
- Forma de acesso aos dados - Se a arquitetura apresenta alguma forma particular de acesso aos dados; e
- Segurança - Se a arquitetura apresenta metodologias para a segurança dos dados, promovendo o controle do acesso ao *data warehouse* e aos provedores de informação, dos usuários de SSD e das transferências dos dados.

Nos próximas seções deste capítulo, para cada arquitetura vista anteriormente, será apresentado um quadro contendo a lista das principais características do referido trabalho com relação aos critérios acima especificados. As informações contidas nestes quadros serão preenchidas de acordo com a seguinte terminologia apresentada por Ciferri [CIFE02]:

- Descrição da Funcionalidade - Caso a arquitetura já ofereça alguma funcionalidade com relação ao critério analisado, será apresentada uma breve descrição desta funcionalidade;

- Não Apresenta - Caso o objetivo da arquitetura esteja diretamente relacionada com o critério analisado, mas nenhuma metodologia e/ou algoritmo sejam especificados ou nenhuma discussão sobre o assunto realizada;
- Apenas indica necessidade - Caso a arquitetura cite a funcionalidade referente ao critério que está sendo analisado, mas não apresenta solução; e
- Não especificado - Caso a arquitetura não considere questões relacionadas ao critério analisado.

4.1 Análise da Arquitetura de Data Warehouse Distribuído de Inmon

O Quadro 4.1 identifica as funcionalidades oferecidas pela arquitetura de *data warehouse* distribuído de Inmon [INM02], em relação aos critérios abordados no início do capítulo.

QUADRO 4.1 – FUNCIONALIDADES OFERECIDAS PELA ARQUITETURA DE DATA WAREHOUSE DISTRIBUÍDO DE INMON [INM02; CIFE02]

Critério	Arquitetura de Inmon [INM02]
Aspectos para fragmentação	<i>Fragmentação por regiões geográficas diferentes ou comunidades técnicas distintas</i>
Aspectos para alocação e/ou replicação	<i>Arquitetura básica: não existe replicação</i> <i>Variação da arquitetura: replicação dos dados locais a serem transferidos para o data warehouse global</i>
Metodologia e/ou algoritmos para fragmentação	<i>Não apresenta</i>
Metodologia e/ou algoritmos para alocação e/ou replicação	<i>Não apresenta</i>
Redirecionamento de consultas processadas de forma centralizada	<i>Apenas indica a necessidade</i>
Gerenciamento de consultas processadas de forma distribuída	<i>Apenas indica necessidade</i>
Controle local aos sites	<i>Apenas indica a necessidade</i>
Controle global aos sites	<i>Apenas indica a necessidade</i>
Manutenção da consistência	<i>Não especificado</i>
Forma de acesso aos dados	<i>Não especificado</i>
Segurança	<i>Não especificado</i>

Na arquitetura de Inmon [INM02] a região geográfica e a comunidade técnica são fatores que podem servir de base para a fragmentação, porém nenhuma metodologia e/ou algoritmo são apresentados para a fragmentação. Na arquitetura básica de Inmon não existe replicação, já na variação desta arquitetura pode haver replicação dos dados armazenados nas áreas de armazenamento do *data warehouse* global, caso a política adotada pela

empresa seja a não remoção destes dados após o envio ao *data warehouse* global. Porém, nenhuma metodologia e/ou algoritmos são apresentados para a alocação e/ou replicação.

Inmon, em sua arquitetura de *data warehouse* distribuído, apenas indica a necessidade de se ter: um redirecionamento das requisições dos usuários de SSD, um gerenciamento destas requisições de forma a melhorar a performance das mesmas, um controle local e global, para que, em princípio, os dados locais sejam acessados somente localmente e os dados globais sejam acessados somente globalmente.

Apesar de poder existir replicação dos dados, como pôde ser visto anteriormente, Inmon não se preocupa com a manutenção da consistência dos dados devido a tal replicação. Os critérios *forma de acesso aos dados* e *segurança* não são abordados no trabalho de Inmon.

4.2 Análise das Arquiteturas de Data Warehousing Distribuído de Moeller

O Quadro 4.2 identifica as funcionalidades oferecidas pelas arquiteturas de *data warehousing* distribuído homogêneo, heterogêneo e com SGBD distribuído único propostas por Moeller [MOE01], de acordo com os critérios listados no início deste capítulo.

QUADRO 4.2 – FUNCIONALIDADES OFERECIDAS PELAS ARQUITETURAS DE DATA WAREHOUSING
DISTRIBUÍDO DE MOELLER [MOE01; CIFE02]

Critério	Arquitetura de Moeller [MOE01]
Aspectos para fragmentação	<i>Passos 7, 9, 10 e 12 (primeira parte da abordagem para o problema da localização dos dados)</i>
Aspectos para alocação e/ou replicação	<i>Passos 7, 8, 9, 10, 11 e 12 (primeira parte da abordagem para o problema da localização dos dados)</i>
Metodologia e/ou algoritmos para fragmentação	<i>Não apresenta</i>
Metodologia e/ou algoritmos para alocação e/ou replicação	<i>Não apresenta</i>
Redirecionamento de consultas processadas de forma centralizada	<i>Funcionalidades oferecidas pela ferramenta de gerenciamento do banco de dados distribuído ou pelo SGBD distribuído</i>
Gerenciamento de consultas processadas de forma distribuída	
Controle local aos sites	<i>Usuários de SSD podem submeter consultas diretamente aos data marts individuais nas arquiteturas de data warehousing distribuído homogêneo e heterogêneo</i>
Controle global aos sites	<i>Funcionalidades oferecidas pela ferramenta de gerenciamento do banco de dados distribuído ou pelo SGBD distribuído</i>
Manutenção da consistência	<i>Discussão de mecanismos tradicionais de controle de concorrência</i>
Forma de acesso aos dados	<i>Não especificado</i>
Segurança	<i>Sugere o uso de ferramentas que melhor se adequem à necessidade da empresa</i>

Moeller [MOE01] apenas discute alguns direcionamentos que podem ser utilizados como base em questões relacionadas à fragmentação, à replicação e à alocação dos dados, como pode ser visto na seção 3.4.4 deste trabalho. Tais direcionamentos não enfocam características intrínsecas de ambientes de *data warehousing*, como exemplo a organização dos dados em diferentes níveis de agregação. Além disto, nenhuma metodologia e/ou algoritmos de fragmentação, alocação e replicação são apresentados no trabalho de Moeller.

As funcionalidades de redirecionamento de consultas processadas de forma centralizada e o gerenciamento de consultas processadas de forma distribuída são oferecidos pela ferramenta de gerenciamento do banco de dados distribuído ou pelo SGBD distribuído, respectivamente nas arquiteturas homogênea ou heterogênea, ou na com SGBD distribuído único.

Os usuários de SSD podem submeter consultas diretamente aos *data marts* individuais nas arquiteturas de *data warehousing* distribuído homogêneo e heterogêneo de Moeller [MOE01], já que os *data marts* locais são autônomos: possuem sua própria

estrutura e conteúdo dos dados, são alimentados por sistemas particulares e não existe coordenação entre os dados ou de processamento entre os *sites*. Já o controle global aos *sites* é feito através da ferramenta de gerenciamento do banco de dados distribuído ou pelo SGBD distribuído.

Em seu livro, Moeller [MOE01] discute a funcionalidade de manutenção da consistência dos dados, exemplificando vantagens e desvantagens de mecanismos de controle de concorrência. Tal discussão engloba mecanismos tradicionais comumente conhecidos, tais como: métodos baseados em bloqueio, métodos baseados em marcadores de tempo (*timestamping*) e métodos otimistas.

O critério forma de acesso aos dados não é especificado na arquitetura de Moeller, enquanto que o critério segurança é tratado de forma detalhada, onde o autor sugere a utilização de *firewalls*, autenticação dos usuários, criptografia e medidas de segurança utilizadas nos SGBD.

4.3 Análise da Arquitetura de Data Warehouse Distribuído Hierárquico de Zhou et al.

O Quadro 4.3 identifica as funcionalidades oferecidas pela arquitetura de *data warehouse* distribuído hierárquico de Zhou et al. [ZZTH00], em relação aos critérios abordados no início do capítulo.

QUADRO 4.3 – FUNCIONALIDADES OFERECIDAS PELA ARQUITETURA DE DATA WAREHOUSE
DISTRIBUÍDO HIERÁRQUICO DE ZHOU ET AL [ZZTH00]

Critério	Arquitetura do HDDW [ZZTH00]
Aspectos para fragmentação	<i>Fragmentação por níveis de hierarquia de relacionamento entre os atributos de uma dimensão</i>
Aspectos para alocação e/ou replicação	<i>Replicação dos dados detalhados de um nível até o topo da hierarquia</i>
Metodologia e/ou algoritmos para fragmentação	<i>Não apresenta</i>
Metodologia e/ou algoritmos para alocação e/ou replicação	<i>Não apresenta</i>
Redirecionamento de consultas processadas de forma centralizada	<i>Não especificado</i>
Gerenciamento de consultas processadas de forma distribuída	<i>Middleware OLAP Distribuído</i>
Controle local aos sites	<i>Não especificado</i>
Controle global aos sites	<i>Middleware OLAP Distribuído</i>
Manutenção da consistência	<i>A manutenção é feita em todos os níveis da arquitetura que contenha referência para o dado alterado</i>
Forma de acesso aos dados	<i>Arquitetura de três camadas: cliente, servidor OLAP local (middleware) e servidor de banco de dados</i>
Segurança	<i>Gerenciamento dos Usuários</i>

Na arquitetura de *data warehouse* distribuído hierárquico de Zhou et al. [ZZTH00], os níveis de hierarquia de relacionamento entre os atributos de uma dimensão são fatores que servem de base para a fragmentação, porém nenhuma metodologia e/ou algoritmo são apresentados. A replicação dos dados é observada à medida que os dados detalhados de um nível mais baixo são integrados em um nível mais alto. Porém, nenhuma metodologia e/ou algoritmos são apresentados para a alocação e/ou replicação.

Na arquitetura de *data warehouse* distribuído hierárquico, através do middleware OLAP Distribuído é possível o gerenciamento das consultas de forma distribuída, como pode ser visto na seção 3.5.1 deste trabalho, enquanto que o redirecionamento de consultas processadas de forma centralizada não é especificado. O controle global aos *sites* é realizado pelo middleware OLAP distribuído e o controle local não é especificado.

A manutenção da consistência dos dados é apenas resumida por Zhou et. al., na necessidade de se manterem consistentes as referências dos dados replicados em todos os níveis da arquitetura, porém neste método, nenhum componente é especificado para tratar este critério.

Finalmente, o critério forma de acesso aos dados é baseado na arquitetura de três camadas e o critério segurança é tratado somente pelo gerenciamento de usuários, onde o usuário de um determinado nível de hierarquia só poderá acessar dados do seu nível para o mais baixo nível da arquitetura.

4.4 Análise do Sistema WebD²W de Ciferri

O Quadro 4.4 identifica as funcionalidades oferecidas pelo sistema WebD²W de Ciferri [CIFE02], de acordo com os critérios listados no início deste capítulo.

QUADRO 4.4 – FUNCIONALIDADES OFERECIDAS PELO SISTEMA WEBD²W [CIFE02]

Critério	Sistema WebD ² W [CIFE02]
Aspectos para fragmentação	<i>Definição das restrições a serem aplicadas à fragmentação identificadas no módulo requisitos do componente de distribuição.</i>
Aspectos para alocação e/ou replicação	<i>Definição das restrições a serem aplicadas à alocação e/ou replicação identificadas no módulo requisitos do componente de distribuição.</i>
Metodologia e/ou algoritmos para fragmentação	<i>Metodologia e/ou algoritmos para fragmentação são inseridos no módulo de fragmentação do componente de distribuição</i>
Metodologia e/ou algoritmos para alocação e/ou replicação	<i>Metodologia e/ou algoritmos para alocação e/ou replicação são inseridos no módulo alocação do componente de distribuição</i>
Redirecionamento de consultas processadas de forma centralizada	<i>Funcionalidades oferecidas pelo componente de consulta do ambiente distribuído</i>
Gerenciamento de consultas processadas de forma distribuída	
Controle local aos sites	
Controle global aos sites	
Manutenção da consistência	<i>Funcionalidade oferecida pelo componente de manutenção do ambiente distribuído</i>
Forma de acesso aos dados	<i>Arquitetura genérica de três camadas para a Web (apresentação, regras de negócio e gerenciamento de dados)</i>
Segurança	<i>Sugere o uso de alguns mecanismos de segurança, como: criptografia, certificação digital, firewalls e programas antivírus.</i>

A identificação dos requisitos para o processo de fragmentação, alocação e replicação dependem tanto de informações sobre o ambiente de *data warehousing* já existente, as quais são obtidas a partir do repositório de metadados, quanto de informações relacionadas às necessidades de distribuição da aplicação, as quais são obtidas a partir de informações externas. Estes requisitos são definidos a partir destas informações e estão presentes no módulo requisitos do componente de distribuição.

Propostas de metodologias e/ou algoritmos para fragmentação são inseridas no módulo fragmentação do componente de distribuição, de acordo com os requisitos para o processo de fragmentação definidos pelo módulo de requisitos. Enquanto que, propostas de

metodologias e/ou algoritmos para alocação e/ou replicação são inseridas no módulo alocação do componente de distribuição, de acordo com os requisitos para o processo de alocação definidos pelo módulo de requisitos e o esquema de fragmentação, que especifica logicamente como os fragmentos do *data warehouse* global podem ser obtidos.

O componente de consulta do ambiente distribuído possui duas funcionalidades mutuamente exclusivas. A primeira delas é denominada de redirecionamento de consultas processadas de forma centralizada. Ou seja, dada uma consulta OLAP, esta funcionalidade redireciona esta consulta ao *site* mais apropriado em respondê-la, para que a consulta seja processada nesse *site* de forma centralizada. A segunda funcionalidade do componente de consulta do ambiente distribuído é denominada gerenciamento de consultas processadas de forma distribuída. Ou seja, dada uma consulta OLAP, esta funcionalidade decompõe esta consulta em diversas subconsultas, de forma que cada uma dessas subconsultas acesse diferentes partes do *data warehouse* distribuído e seja enviada e processada ao *site* mais apropriado em respondê-la. Os resultados parciais das subconsultas são agrupados até a obtenção de um resultado final que será retornado para o usuário [CIFE02].

As funcionalidades oferecidas pelo componente de consulta do ambiente distribuído do sistema WebD²W podem ser enfocadas sob duas diferentes perspectivas. A primeira perspectiva está relacionada ao acesso local a cada *site* que participa do ambiente distribuído, enquanto que a segunda perspectiva diz respeito ao acesso global ao ambiente distribuído como um todo. No sistema WebD²W, essas duas perspectivas são nomeadas, respectivamente, de funcionalidade de acesso local e de funcionalidade de acesso global e foram detalhadas na seção 3.6.6 deste trabalho.

O componente de manutenção do ambiente distribuído é responsável por manter os dados consistentes em toda a arquitetura, ou seja, no *data warehouse* global e nas unidades de distribuição. Segundo Ciferri [CIFE02], este componente pode desempenhar duas funcionalidades diretamente relacionadas, que são: a manutenção da consistência intra-site e a entresites, que foram vistas em maiores detalhes na seção 3.6.4 deste trabalho.

O principal motivo da escolha da arquitetura de três camadas como forma de acesso aos dados do *data warehouse* distribuído no sistema WebD²W refere-se ao fato de que esta forma de organização torna possível a substituição de qualquer um dos componentes lógicos da aplicação (apresentação, regras de negócio ou gerenciamento de dados) sem que seja necessário modificar os demais componentes. Garante, desta forma, a independência da aplicação no acesso ao *data warehouse*.

A tese de Ciferri [CIFE02] apesar de não propor tratar o problema da segurança, sugere a utilização de mecanismos de segurança para tornar a arquitetura mais segura, como por exemplo: criptografia, certificação digital, *firewalls* e programas antivírus.

4.5 Benefícios e Desvantagens das arquiteturas abordadas

Nesta seção, serão apresentados, individualmente, os principais benefícios e desvantagens de cada uma das arquiteturas abordadas anteriormente.

Na arquitetura proposta por Inmon [INM02], pode-se apresentar como vantagens:

- O fato de se ter um *data warehouse* global materializado, que armazena todos os dados agregados, onde não é necessário o gerenciamento de consultas OLAP distribuídas, aumentando a performance no momento da consulta; e

- O armazenamento dos dados agregados somente na sede da empresa para dá suporte à decisão e a divisão dos *data warehouses* locais em regiões geográficas diferentes contribuem face à realidade da globalização.

Já os pontos negativos desta arquitetura se baseiam nos seguintes fatos:

- Nenhum tipo de mecanismo de segurança é especificado; e
- Não apresentar mecanismos para a manutenção da consistência dos dados, apesar de poder existir replicação dos dados.

Na arquitetura proposta por Moeller [MOE01], pode-se listar como vantagens:

- Apresentação de uma abordagem prática para o problema da localização dos dados, de forma a maximizar o desempenho das consultas dos usuários e minimizar o tempo gasto no carregamento dos dados;
- Discussão de mecanismos tradicionais de controle de concorrência;
- Sugere a utilização de ferramentas para o aumento da segurança na arquitetura; e
- *Data Marts* autônomos entre si, nas arquiteturas de *data warehousing* distribuído homogêneo e heterogêneo, facilitando manutenção e o acesso local.

As desvantagens desta arquitetura podem ser analisadas através dos seguintes aspectos:

- O fato de se ter um *data warehouse* virtual pode ser considerada uma desvantagem em ambientes de *data warehousing* distribuído, pois nestes ambientes o desempenho das consultas é mais importante do que a consistência imediata dos dados junto aos *data warehouses* locais; e
- Pouco detalhamento da ferramenta de gerenciamento banco de dados distribuídos, que é o componente mais importante desta arquitetura.

Na arquitetura proposta por Zhou et. al. [ZZTH00], pode-se apresentar como vantagens:

- O detalhamento do middleware OLAP distribuído, para que se possa gerenciar consultas OLAP distribuídas, com transparência de fragmentação, alocação e replicação para o usuário; e
- Implementação e manutenção da arquitetura é fácil, devido aos *data marts* serem praticamente independentes entre si.

Já os pontos negativos desta arquitetura se baseiam nos seguintes fatos:

- A questão da segurança é apresentada de modo superficial; e
- Inexistência de um componente que se encarrega de fazer a manutenção da consistência dos dados.

Na arquitetura proposta por Ciferri [CIFE02], pode-se listar como vantagens:

- A definição das funcionalidades de cada componente e as responsabilidades dos usuários na arquitetura;
- Utilização de algoritmos de fragmentação horizontal e vertical para a distribuição dos dados nas diversas unidades de distribuição; e

- Permite o acesso de usuários de *sites* que não armazenam porções do *data warehouse* distribuído.

As desvantagens desta arquitetura podem ser analisadas através dos seguintes aspectos:

- Utilização da abordagem *top-down* de desenvolvimento, visto que, a abordagem *bottom-up*, atualmente, é a mais usada nas empresas devido a implementação e retorno mais rápido.

4.6 Recomendações para a construção de arquiteturas de data warehousing distribuído

A partir da análise desenvolvida neste trabalho sobre as arquiteturas de *data warehousing* distribuídos, foram definidas recomendações para a construção de outras arquiteturas que também tratem o problema da distribuição do *data warehouse*. Assim, esta seção sugere algumas boas práticas para o desenvolvimento destas arquiteturas.

Primeiramente, é preciso definir qual a abordagem de desenvolvimento que será utilizada: a *bottom-up* ou a *top-down*. Por causa disto, serão feitas sugestões específicas para cada uma destas abordagens.

Para a abordagem de desenvolvimento *bottom-up* de uma arquitetura de *data warehousing* distribuído, as recomendações para a sua construção são:

- Utilizar uma ferramenta cliente que possibilite a geração de relatórios e a interação do usuário com o ambiente distribuído, a partir de consultas no formato MDX (*Multidimensional Expressions*), que é o mais utilizado atualmente para consultas multidimensionais;
- Utilizar uma ferramenta ETL em cada *site* de distribuição para fazer a extração, transformação e carga dos sistemas operacionais para o *data warehouse*;
- Utilizar um middleware OLAP distribuído ou definir um componente de consulta do ambiente distribuído, que proporcione o redirecionamento, o gerenciamento e a posterior integração das consultas OLAP feitas pelos usuários de SSD;
- Se os dados forem replicados nos *sites*, será necessária a definição de um componente de manutenção do ambiente distribuído, que manterá os dados replicados consistentes;
- Definir um gerenciador de metadados para manipular e um repositório de metadados para armazenar informações semânticas sobre os esquemas dos *data warehouses* locais e do *data warehouse* global, onde estes dois componentes terão que ser replicados em cada *site* do ambiente distribuído;
- Utilizar *firewalls*, autenticação de usuários, gerenciamento de usuários, criptografia e programa antivírus para aumentar a segurança dos dados na arquitetura;
- Utilização da arquitetura genérica de três camadas da Web como forma de acesso aos dados da arquitetura de *data warehouse* distribuído, com o objetivo de facilitar a manutenção de qualquer um dos componentes lógicos da aplicação (apresentação, regras de negócio e gerenciamento de dados);

- Utilizar a abordagem prática para o problema da localização dos dados proposta por Moeller [MOE01] para maximizar o desempenho do sistema no atendimento às consultas e minimizar o tempo gasto no carregamento dos dados; e
- Utilizar um SGBD confiável, robusto, com baixos tempos de repostas e tolerante a falhas para o armazenamento de todas as informações.

Na abordagem *bottom-up* os *data marts* são construídos incrementalmente, sugerindo que os dados já foram previamente fragmentados e alocados de acordo com o contexto dos *data marts* locais, o que ocasiona a não utilização do componente de distribuição nesta abordagem, que seria responsável por fragmentar, alocar e carregar os dados nos *data marts* locais a serem distribuídos. Já na abordagem *top-down* este componente é indispensável, pois é através dele que serão gerados os fragmentos a serem alocados, replicados e carregados em cada unidade de distribuição.

Com isso, pode-se observar que, todas as recomendações feitas anteriormente para a construção de uma arquitetura de *data warehousing* distribuído utilizando a abordagem *bottom-up* de desenvolvimento, também podem ser usadas para a abordagem *top-down*. A única diferença é a definição do componente de distribuição na arquitetura com abordagem *top-down* de desenvolvimento, que indicará algumas recomendações extras, listadas abaixo:

- Identificar conjunto de critérios que deve ser utilizado para a definição de restrições a serem aplicadas aos processos de fragmentação, replicação e de alocação dos dados do *data warehouse* global;
- Utilizar metodologias e algoritmos para a fragmentação horizontal, vertical ou mista dos dados do *data warehouse* global;
- Utilizar metodologias e algoritmos baseados em heurísticas para alocação e/ou replicação dos fragmentos do *data warehouse* global em diferentes *sites*; e
- Definir um componente responsável pelo carregamento dos dados do *data warehouse* global nas unidades de distribuição, com base no esquema de fragmentação e no esquema de alocação.

5. Conclusões e Trabalhos Futuros

Atualmente, com o grande volume de dados nas empresas, o ambiente de *data warehousing* mostra-se como uma interessante tecnologia para auxiliar à tomada de decisões estratégicas das empresas de forma rápida e segura.

Entretanto, devido ao crescente volume de dados manipulados, aumento do número de consultas em função do crescimento de usuários e às dispersões geográficas, os ambientes de *data warehousing* podem estar perdendo em desempenho e disponibilidade.

Com isso, surgiram propostas de arquiteturas que tentam solucionar o problema da distribuição em ambientes de *data warehousing*, com o intuito de aumentar a performance e disponibilidade dos mesmos.

Neste trabalho, foram abordadas quatro destas propostas de arquiteturas de ambiente de *data warehousing* distribuído, respectivamente as propostas feitas por Inmon [INM02], Moeller [MOE01], Zhou et al. [ZZTH00] e Ciferri [CIFE02], para uma análise comparativa entre as mesmas, baseada em critérios abordados neste trabalho, destacando as vantagens e desvantagens de cada uma destas arquiteturas.

O trabalho finalizou com a apresentação de recomendações para construção de arquiteturas de distribuição de *data warehouse*, considerando tanto a abordagem *bottom-up*, quanto a *top-down*, a partir dos resultados da análise desenvolvida.

5.1 Trabalhos Futuros

Em continuidade ao trabalho desta pesquisa, recomenda-se como trabalho futuro o desenvolvimento de ferramentas que dêem suporte a ambientes de *data warehousing* distribuído, obtendo todas as vantagens proporcionadas pela distribuição. A análise comparativa destas arquiteturas, descrita neste trabalho, contribui para facilitar a escolha da melhor arquitetura que servirá de base para o desenvolvimento destas ferramentas.

No desenvolvimento das referidas ferramentas, pode-se utilizar as recomendações feitas neste trabalho como base, considerando:

- O software Kettle [PEN07] como ferramenta ETL para o processo de extração, tradução e carga dos dados dos ambientes operacionais para os *data warehouses* locais ou o *data warehouse* global;
- O SGBD Oracle [ORA07] como ferramenta responsável por armazenar todas as informações do ambiente distribuído;
- Uma extensão do software Mondrian [PEN07] para possibilitar o gerenciamento, o redirecionamento e a posterior integração das consultas OLAP feitas pelo usuário de SSD ao ambiente distribuído, funcionando como um Middleware OLAP Distribuído;
- O software JFreeReport [PEN07] como ferramenta responsável por gerar relatórios para a alta gerência da empresa;
- Java [SUN07] como linguagem de programação para desenvolver o componente de manutenção, o gerenciador de metadados e, caso seja necessário, o componente de distribuição; e

- O software Panda Antivirus + *Firewall* 2007 [PAN07] como ferramenta responsável por aumentar a segurança da rede e dos computadores que fazem parte da mesma.

Essas ferramentas, que darão suporte a ambientes de *data warehousing* distribuído, terão como objetivo principal aumentar a disponibilidade dos dados, como também, o acesso a estes dados.

Referências Bibliográficas:

- [AJLW98] ALBRECHT, J.; LEHNER, W. 1998. On-Line Analytical Processing in Distributed Data Warehouses. In *Proceedings of the 1998 international Symposium on Database Engineering & Applications* (July 08 - 10, 1998). IDEAS. IEEE Computer Society, Washington, DC, 78.
- [BAR96] BARQUINI, Ramon; *Planning and designing the Warehouse*, New Jersey, Prentice-Hall, 1996, 311 pg.
- [CCSF02] CIFERRI, C.D.A., SOUZA, F.F. Focusing on Data Distribution in the WebD2W System. In *Proceedings of the 4th International Conference on Data Warehousing and Knowledge Discovery*, pages 265-274, Aix-en-Provence, France, September 2002, volume 2454 of *Lecture Notes in Computer Science*, Springer, 2002.
- [CHAU97] CHAUDHURI, S.; DAYAL, U. *An Overview of Data Warehousing and OLAP technology*. SIGMOD record, 26(1):65-74, 1997.
- [CIFE02] CIFERRI, C. D. A.; *Distribuição dos Dados em Ambientes de Data Warehousing: O Sistema WebD2W e Algoritmos Voltados à Fragmentação Horizontal dos Dados*. Universidade Federal de Pernambuco, 2002.
- [COSTA05] COSTA, M.V.C.S., *Fragmentação Vertical de Data Warehouse em Termos de Medidas Numéricas: Um Algoritmo Básico*. Universidade Federal de Pernambuco, 2005.
- [DAL99] DAL'ALBA, Adriano. *Um estudo sobre Data Warehouse*, Faculdade de Ciências Administrativas Valinhos, 1999
- [ERNS04] ELMASRI, Ramez; NAVATHE, Shmakant B.; *Fundamentals of Database Systems*, 4ª ed., Addison-Wesley, 2004.
- [FFRC07] FAVARETO, F. ; Rhoden, C.A. *Considerações sobre Atividades de Identificação, Localização e Tratamento de Dados na construção de um Data Warehouse*, Pontifícia Universidade Católica do Paraná . Disponível em www.pr.gov.br/congressobd/Artigos/Artigo_05.pdf, acessada em 23/03/2007.
- [INM96] INMON, W. H. – *Building the Data Warehouse*, John Wiley & Sons Inc., New York, 1996.
- [INM02] INMON, W. H. – *Building the Data Warehouse*, John Wiley & Sons Inc., New York, 2002.
- [KIM96] KIMBALL, Ralph; *The Data Warehouse Toolkit*. John Wiley & Sons Inc., New York, 1996.

- [MOE01] MOELLER, R.A. *Distributed Data Warehousing using Web Technology*. AMACON–American Management Association, USA, 2001.383 pp. ISBN 0-8144-0588-6.
- [NASC05] NASCIMENTO, A.L, *Um Algoritmo baseado em Grafo de Derivação para Realizar Fragmentação Vertical Unidimensional em Data Warehouse*.Universidade Federal de Pernambuco, 2005.
- [NOA00] NOAMAN, A. Y., *Distributed Data Warehouse Architecture And Design*. University of Manitoba. Canada, 2000.
- [ORA07] ORACLE CORPORATION. Disponível em: www.oracle.com, acessado em 22/03/2007.
- [PAN07] PANDA SOFTWARE. Disponível em: <http://www.pandasoftware.com>, acessado em 22/03/2007.
- [PEN07] PENTAHO OPEN SOURCE BUSINESS INTELLIGENCE. Disponível em: <http://www.pentaho.com>, acessado em 22/03/2007.
- [SOAR98] SOARES, V.J.A, *Modelagem Incremental no Ambiente de Data Warehouse*. Universidade Federal do Rio de Janeiro, 1998.
- [SUN07] SUN MICROSYSTEMS. Disponível em: <http://java.sun.com>, acessado em 22/03/2007.
- [TVFR07] TIMES, Valéria; FIDALGO,Robson. *DW, OLAP e Modelagem Multidimensional*. Disponível em: www.cin.ufpe.br/~if695, acessado em 21/01/2007.
- [TERRA07] Business Intelligence. Disponível em: <http://paginas.terra.com.br/negocios/processos2002/bi.htm>, acessado em 03/04/2007.
- [ZZTH00] ZHOU, Shuigeng; ZHOU, Aoying; TAO, Xiaopeng; HU, Yunfa. *Hierarchically Distributed Data Warehouse, hpc*, p. 848, The Fourth International Conference on High-Performance Computing in the Asia-Pacific Region-Volume 2, 2000.
- [WDCP96] WELLS, D. CARNELLY, P. *Ovum eveluation: the Data Warehouses*. Ovum Ltd., Londo, 1996.