Estudo de Algoritmos e Técnicas de manutenção aplicadas a projetos de Data Warehouse

Clemente A. Lana[1], Giglio, G. P. Morais[2]

Centro de Ensino Superior de Juiz de Fora (CES/JF)

Rua Halfeld 1179 – 36.016-000 – Juiz de Fora – MG – Brasil

andre.lana@hotmail.com.br, giuprado@gmail.com

Abstract. *Once the Data Warehouse (DW) construction is defined, the term maintenance is often left aside, which may cause future problems as we talk about a huge and systemic structure being necessary the use of specific maintenance tools, that if is neglected can put the storage data in risk. To minimize the need of correct a formed problem, the Data Warehouse maintenance factor should be covered since the project beginning, and should be supported by many management plans as requirements, architecture, configuration so that be properly planned. This article analyzes which maintenance measures must be applied in a DW with the intuit that the same ones by the life cycle keep attending the original purpose for which were developed. This analysis is based on algorithms and specific techniques for a consistent and robust maintenance, were we intend to write some conclusions.*

Resumo. *Definida a construção de um Data Warehouse (DW), o termo manutenção muitas vezes é deixado de lado, o que pode acarretar problemas futuros, uma vez em que falamos de uma estrutura grande e sistêmica, onde se faz necessário a utilização de ferramentas específicas, e se negligenciado, pode pôr em risco as informações armazenadas. O fator manutenção nos Data Warehouses deve ser abrangido logo do início de seu projeto, minimizando a necessidade de agir reativamente para combater um problema já formado. Deve ser apoiada por diversos planos de gerência, tais como gerência de requisitos, de arquitetura e de configuração, para que desta forma a mesma seja devidamente planejada. Este trabalho visa analisar quais medidas devem ser tomadas ao aplicar a manutenção em um DW, com o intuito de que os mesmos, ao longo do seu ciclo de vida, continuem atendendo o propósito pelo qual foram desenvolvidos. Esta análise será baseada em algoritmos e técnicas específicas para que haja uma manutenção consistente e robusta, onde se pretende tecer algumas considerações sobre as mesmas.*

1. Introdução

Em um mercado corporativo cada vez mais competitivo, se faz necessário a constante busca por vantagens e diferenciais. (Carvalho, 2009).

Em meio a necessidade de informações organizacionais, crescente nos anos 90, o conceito *Data Warehouse* (DW) é apresentado como solução empresarial, possibilitando extrair informações gerenciais para tomada de decisões (Javed e Rafique, 2006).

Para auxiliar nas decisões estratégicas empresariais, o DW coleta informações de diversos bancos de dados, e as mantém em um único local com o propósito de otimizar as consultas (Javed e Rafique, 2006).

Segundo o estudo de Ferreira (2002), pode-se dizer que, a riqueza de uma empresa não está no volume de dados armazenados, nem em informações geradas por sistemas de aplicação comercial, mas pelo conhecimento da informação (Ferreira, 2002).

Sistemas de apoio a decisão auxiliam em situações em que o julgamento humano age como uma contribuição importante ao processo de resolução, no entanto, a limitação humana para processar tais informações atrapalha (Ferreira, 2002).

O DW pode ser definido como um tipo de SAD (Sistema de Apoio a Decisão), sendo uma fonte de consultas de um empreendimento (Ferreira, 2002).

O objetivo primário de um DW é fornecer recursos que possibilitem a transformação de uma base de dados *on-line* OLTP (*On-Line Transaction Processing*) para uma base de dados maior OLAP (*On-Line Analitic Process*) que possua estrutura para armazenar todos os dados essenciais de uma organização (Ferreira, 2002).

Ao término dos anos 90, o avanço da internet alterou toda a estrutura de organização das informações empresariais. Grande parte das organizações migrou informações existentes nas estruturas transacionais para os modelos web. Segundo os mesmos, várias empresas atuantes no desenvolvimento dos DW afirmavam que a mudança na estrutura seria algo ágil de se realizar, porém a realidade foi diferente. Poucos conseguiram realizar a atualização necessária para continuar a atender a demanda crescente, entretanto os usuários continuavam insatisfeitos, reclamando da qualidade do DW (Javed e Rafique, 2006).

Além das manutenções realizadas para manter o DW sempre atualizado durante o ciclo de vida, existe a importância de se realizar a manutenção dos dados armazenados. De uma maneira geral, para que sejam otimizadas as informações no DW, deve ser dispensado algum tempo a fim de que futuramente, quando for necessário tomar alguma decisão gerencial, os mesmos possam ser encontrados com facilidade.

A grande utilização do DW tanto durante seu ciclo de vida, como com a quantidade de informações inseridas, pode fazer com que o mesmo apresente dificuldades ao realizar as extrações de dados como também apresentá-los defasados. Portanto, no início do projeto de sua estrutura deve ser planejada também uma forma de aplicar a manutenção estrutural e dos dados no decorrer do ciclo de vida.

O objetivo deste estudo visa analisar quais medidas devem ser tomadas ao aplicar a manutenção em um DW, com o intuito de continuar atendendo ao propósito pelo qual foi desenvolvido, ao longo do seu ciclo de vida.

O trabalho possui também o objetivo de apresentar o assunto DW através de um levantamento bibliográfico, utilizando-se de trabalhos mais recentes, se comparados com os livros disponíveis sobre o assunto, a fim de ser mais uma contribuição aos estudantes de interesse e profissionais envolvidos com essa tecnologia.

Este trabalho foi dividido, inicialmente sendo descritas as características da estrutura de um DW na unidade 2, onde também é apresentado um estudo de caso de uma implantação DW para apoio de decisões. Na unidade 3 são levantados os tipos e técnicas de manutenção aplicadas em um ambiente DW e, na unidade seguinte, são levantadas características dos processos e técnicas de manutenção abordadas, apontando resultados encontrados em estudos de casos realizando comparativos que objetivam demonstrar os principais aspectos e resultados encontrados. Na última unidade, se

apresenta uma comparação entre resultados encontrados nos estudos de caso, levantando pontos positivos e negativos da utilização dos mesmos, passando às considerações finais do presente trabalho.

2. Construção e Manipulação de um Data Warehouse

No início dos anos 90, com o propósito de auxiliar e suprir a demanda das empresas na tomada de decisões gerenciais, surgiram os *Data Warehousing* (Chaudhuri e Dayal, 1997). Um dos aspectos mais importantes do DW é o fato de ser integrado. Permitindo que os dados sejam repassados para o DW, diretamente do ambiente operacional. Desde então, o sucesso da solução pode ser comprovado pelo aumento dos investimentos em projetos DW ao redor do mundo (Ferreira, 2002).

O DW se tornou reconhecido por coletar dados de diversos sistema fonte de dados, padronizá-los e gerar informações que auxiliem a tomada de decisões gerenciais (Javed e Rafique, 2006).

2.1 Arquitetura

A arquitetura do DW deve ter uma preparação para poder receber os mais variados tipos de dados, disponibilizados pelos sistema fonte, que são responsáveis por fornecer informações transacionais. Entretanto são sistemas que não se tem controle sobre as informações geradas e geralmente não são preparados para executarem consultas da forma como normalmente é feito no DW (Kimball, 2013).

Para que seja possível receber essas informações, os dados são convertidos para um estado uniforme, a fim de permitir a carga no DW, fazendo com que inconsistências provindas da aplicação sejam desfeitas (Ferreira, 2002).

Segundo Dill (2002), os ambientes DW processam grandes quantidades de dados, muitas das vezes não possuindo um padrão definido. Por este motivo, o dimensionamento de carga torna-se complexo para ser efetuado. Além disso, este dimensionamento torna-se mais complexo para ser executado, devido a diferença de utilização dos sistemas DW, quando comparados aos bancos de dados operacionais. No DW os dados são armazenados de forma a manter um histórico das informações da empresa (Dill, 2002).

O dado armazenado no ambiente DW se apresenta apenas sobre a forma de consulta. Diferente de quando se encontram no ambiente OLTP, onde normalmente sofrem *updates* realizadas em operações de inserção, alteração e exclusão, além de consulta.

2.1.1 OLTP e OLAP

No intuito de melhorar o entendimento sobre a forma de processamento dos dados contidos, serão apresentados dois tipos de sistemas, OLTP e OLAP.

Sistemas implementados para OLTP retornam os dados de forma instantânea, sem um tratamento. A importância para o mesmo seria a conclusão da transação realizada durante a consulta.

Em contrapartida, temos os sistemas OLAP, que retornam as informações de maneira a serem aproveitadas na tomada de decisões gerenciais de uma organização. Os

dados são formatados em informações requeridas para a gestão do negócio. Neste caso, temos uma preocupação com a análise dos dados.

Segundo Ferreira, (2002), as atividades relacionadas a consultas e apresentações de dados provenientes de um DW, constituem o processamento analítico *On-Line*. Tais sistemas OLAP auxiliam analistas e gerentes a sintetizarem informações organizacionais através de visões, e análise dos dados em diversas situações.

2.2 Projeto e escopo

O primeiro passo no processo de desenvolvimento de um DW é a definição da estrutura organizacional. Neste processo é definido qual equipe ficará responsável pela tomada de decisões e aprovação dos resultados obtidos, qual equipe exercerá o controle geral do projeto.

Para realização do escopo, devem ser levados em conta os sistemas que proverão os dados para alimentar o DW. No estudo de caso de Santos, Almeida, Tachinardi e Gutierrez (2006), o DW teve como sistema fonte de dados, três sistemas, sendo eles o SIA (Sistema de Informações Ambulatoriais), SIH (Sistema de Informação Hospitalares) e CNES (Cadastro Nacional de Estabelecimentos de Saúde).

Para realizar o levantamento de requisitos, durante o processo de implantação do DW, utilizou-se a técnica "*Source-Driven*", que identifica os requisitos pelos sistemas que fornecem os dados que serão inseridos no DW.

Durante o processo de desenvolvimento, ainda no estudo de caso, para que a expectativa dos usuários fosse atendida, os desenvolvedores optaram para utilizar a implantação incremental, que por sua vez permite a liberação de resultados em um tempo menor.

2.3 Preparação dos dados

O processo o qual os dados passam, para que sejam convertidos é denominado processo de ETL (Extração, Transformação e Carga). Esta etapa permite que os dados extraídos de fontes externas possam ser integrados e transformados antes de estarem aptos para serem carregados no DW. Abaixo, temos os detalhes de cada um dos processos realizados:

- **Extração:** Processo onde são obtidos os dados. Em grande maioria, os dados são coletados de diversas fontes de dados e enviados para a área de transformação, para posteriormente serem trabalhados.
- **Transformação:** Nesta etapa, os dados extraídos de suas fontes são analisados para que então seja possível definir quais atividades deverão ser realizadas, bem como limpeza, eliminação de campos ou dados que não serão úteis ao DW, combinação de fontes de dados quando as mesmas apresentam o mesmo valor, normalização dos dados.
- **Carga:** Neste último passo, são carregados os dados no *Data Warehouse*. Tal função requer checagem da integridade dos dados, otimização do processo de carga e suporte às necessidades do processo de carga, como a eliminação e inclusão de índices (Hokama, Camargo, Fujita e Fogliene, 2004).

2.3.1 Exemplo de aplicação de um Data Warehouse

Para demonstrar a aplicação dos processos de extração, transformação e carga, pode-se ter como exemplo, o estudo de caso de Santos, Almeida, Tachinardi e Gutierrez (2006) descrito a seguir, onde foram feitos levantamentos das situações encontradas, soluções aplicadas durante implantação de um DW na Secretaria de Estado de Saúde de São Paulo para gestão da saúde pública (Santos, Almeida, Tachinardi e Gutierrez, 2006).

Logo ao dar início a implementação do projeto, a estrutura organizacional foi elaborada, criando a hierarquia para coordenar e executar atividades. Neste caso, a equipe de TI ficaria responsável pelo desenvolvimento enquanto o comitê executivo validaria os resultados de acordo com o necessário.

No processo de montagem do escopo foram analisadas as fontes de dados que alimentarão o DW. Devido à grande quantidade de sistemas, foram definidas duas fases de execução dividindo a implementação dos sistemas da saúde, onde primeiramente seriam integrados, em um prazo de seis meses, três sistemas, de acordo com a Figura 1. São eles, um sistema de controle das informações ambulatoriais, o Sistema de Informações Hospitalares - SIH e Cadastro Nacional de Estabelecimento de Saúde - CNES.

Na figura 1 temos uma estrutura simplificada de um DW. Os dados partem de sua origem, representada na imagem pela estrutura "Dados operacionais". Porém antes que cheguem ao DW, precisam passar antes pelo processo de carga ou ETL para posteriormente chegarem ao destino em um modelo dimensional. Por fim, para que seja possível realizar a extração de informações gerenciais, os dados passam por uma ferramenta OLAP que através de uma série de recursos permite a geração das informações. Na estrutura, os "Metadados" tem a função de documentar os processos ETL e OLAP, fornecendo um dicionário de dados.

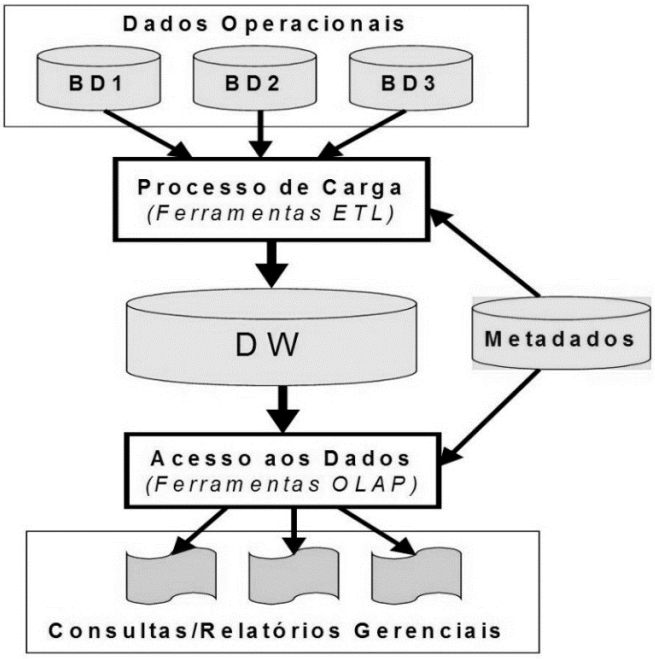

Figura 1. Solução de um Data Warehouse
Fonte: Santos, Almeida, Tachinardi e Gutierrez (2006)

A técnica *Source Driven*, foi utilizada para levantamento de requisitos, identificando-os a partir dos sistemas provedores dos dados. Tal técnica foi adotada devido o escopo amplo do projeto, restrição dos dados fornecidos na origem e a necessidade de gerenciamento desses dados por outra organização.

Devido à necessidade de liberação de subprodutos em um período curto de tempo, definiu-se como modelo do projeto o modelo relacional, e determinou-se a utilização da implantação incremental. O modelo relacional permite um processo ágil para a carga e consulta de dados, e de forma a satisfazer a expectativa do usuário a utilização da implantação incremental permite a apresentação gradual dos resultados do novo produto. Foi determinado também a utilização do modelo dimensional estrela, possuindo o relacionamento de todas as dimensões diretamente com o fato, permitindo adotar uma forma mais simplificada para armazenar o histórico das alterações realizadas.

No processo de definição de ferramentas, optou-se por utilizar produtos Oracle (*Oracle 10g, Oracle IAS, Oracle Warehouse Builder*) para desenvolvimento do DW, enquanto para carga dos dados, além do *Warehouse Builder*, utilizou-se a ferramenta *Compucarga*. Tais ferramentas foram selecionadas para implementar robustez ao projeto, e por apresentarem um custo acessível.

Para o processo de carga dos dados, foram definidos os principais fatos produzidos pelos sistema fonte de dados da primeira fase do projeto. Durante esta etapa, os autores encontraram algumas dificuldades para integrar o tipo de dados gerado por um dos sistemas de cadastro. Por se tratar de sistema para armazenamento de dados cadastrais, a solução aplicada foi permitir a visualização histórica dos dados.

Devido a imprescindibilidade da utilização do modelo relacional para carga e consulta ágil das informações, o processo de carga dos dados foi separado em dois estágios. No primeiro estágio as informações serão repassadas para um espaço de armazenamento nomeado *STAGE*. Somente após passar pelo *STAGE*, os dados serão carregados para o banco dimensional. Nessa primeira etapa, a ferramenta *Compucarga* é utilizada para automatizar o processo de download, versionamento, validação do conteúdo e estrutura do arquivo, garantindo assim a integridade do mesmo. Ainda no primeiro estágio, os dados passam pelo processo de limpeza e padronização, reduzindo a complexidade para o segundo estágio.

Para a carga dos dados do banco de dados dimensional, são utilizados scripts gerados a partir do OWB. Devido aos processos de limpeza, padronização, consistência dos dados já realizados na primeira etapa, o segundo passo se torna menos complexo. Por fim, o usuário tem acesso ao DW através de um portal, que permite visualizar os relatórios gerados.

Ao fim do estudo de caso, foram levantadas dificuldades, como falta de qualidade dos dados, salto tecnológico muito alto, dificuldade no processo de integração dos dados originais, dentre outros. Os dados não possuíam tratamento, e em diversas vezes foram encontradas inconsistências referenciais, duplicidade de informações geradas por cadastros independentes sendo necessária a análise por um especialista. Outra dificuldade, encontrada já em relação aos sistemas de origem dos dados seria o "Salto Tecnológico", uma vez que os sistemas antes do projeto retornavam dados com extensões DBF manipulados pelo Microsoft Excel, ou pelo aplicativo do DATASUS (TABWIN).

3. Técnica de manutenção em um Data Warehouse

Dentre os objetivos de um *Data Warehouse*, o podemos destacar o fornecimento de informações precisas e confiáveis, para tomada de decisões gerenciais (Javed e Rafique, 2006).

Na pesquisa realizada para este trabalho, pode-se observar que estudos apontam, no entanto, que manter o DW pelo ciclo de vida através de manutenções pode ser mais complexo e necessitar de mais recursos do que para o desenvolvimento do sistema. Mesmo que tenha sido entregue com sucesso, o processo de manutenção deve ser realizado constantemente. Caso sejam inseridos dados de baixa qualidade e sem o controle constante de manutenções, o sistema corre o risco de apresentar problemas que poderão levar a falhas comprometendo assim o funcionamento correto do sistema (Schwaickardt, E 2013).

3.1 Manutenção estrutural

Em estudos realizados sobre DW por Bischoff e Alexander (1997), foram facilmente identificados artigos e estudos de caso desenvolvidos a respeito da criação do DW, porém nenhuma referência sobre como realizar a devida manutenção durante o ciclo de vida (Bischoff e Alexander 1997).

Em seu estudo, Javed e Rafique (2006) apontam fatores que precisam ser analisados em DW que já estão em funcionamento. Dentre eles se destacam o crescimento exponencial do tamanho do DW devido à grande quantidade de informações, e consequentemente a quantidade de processamento requerido, além do fator que interliga diretamente os dados aos sistema fonte. Caso os sistema fonte de dados passem por atualizações constantes, o DW deverá ser constantemente analisado para manter principalmente as informações confiáveis e corretas. A dificuldade encontrada no processo de manutenção de DW é justificada pelo fato da construção do DW se mostrar menos complexa do que manter a estrutura do sistema. Estrutura que por sua vez refere-se a atributos, fontes dos dados de informações, dimensões que apoiam a arquitetura do DW (Javed e Rafique 2006).

3.2 Manutenção dos dados no Data Warehouse

Segundo Miranda (2014), *Views* ou Visões são basicamente o resultado de consultas realizadas em um banco de dados, tendo como principal função manter a integridade dos dados originais salvos. As *Materialized Views* (Visões Materializadas), por sua vez, são geradas quando a tupla da Visão é armazenada (Gupta e Mumick 1995). Neste cenário as informações são salvas tornando seu uso vantajoso com relação ao ganho de agilidade nas consultas, principalmente quando contém cálculos matemáticos. No entanto, tal recurso demanda muito espaço, o que justifica o uso amplo em DW (Miranda, 2014).

O DW coleta dados das diversas fontes conectadas durante a integração e pode armazená-las como Visões Materializadas. Portanto para garantir que o processo de decisão seja sempre eficaz, é necessário garantir a consistência dos dados. (Agner, 2000).

Devido às constantes alterações dos dados presentes nos sistemas integrados ao DW, torna-se necessário aplicar manutenção nas informações já armazenadas

anteriormente com o intuito de mantê-las sempre atualizadas. Para que seja possível realizar a manutenção destes dados, existem três técnicas de manutenção, sendo elas o reprocessamento, a replicação e atualização incremental (Agner, 2004).

- **Reprocessamento:** No momento em que as fontes de dados sofrem modificações, a Visão Materializada precisará ser atualizada. Com a técnica de reprocessamento o conteúdo da Visão Materializada é descartado e a visão é materializada novamente com os dados atualizados (Saccol, 2001). Entretanto, durante a atualização da visão, todos os processamentos de consulta devem ser interrompidos (Agner, 2000). Importante ressaltar que, segundo informação disponibilizada no *IMB Knowledge Center*, a Visão Materializada não pode ser deletada e criada novamente. Caso isso venha a ocorrer a ID da Visão também mudará podendo ocasionar falhas nos objetos já referenciados a mesma. (IBM Knowledge Center, 2014).
- **Replicação:** A técnica consiste em manter atualizadas as Visões relacionadas aos dados de origem, sempre que estes forem alterados (Agner, 2004). Apesar de garantir que cada nova consulta realizada, as cópias estarão atualizadas, a desvantagem em sua utilização seria a utilização de espaço para manter atualizados todos os relacionamentos com a Visão modificada (Zhuge, 1995).
- **Manutenção Incremental:** A medida que a atualização é aplicada aos dados na fonte, tornam-se inconsistentes as Visões Materializadas (Ciferri, 2002). Através de algoritmos específicos, é possível atualizar parcialmente a Visão, sem a necessidade de interromper o processamento de consultas ao DW.

3.3 Algoritmos de apoio a manutenção

O ambiente *Data Warehouse* pode possuir diversas fontes de dados, dentre elas sistemas legados. Tais sistemas podem informar ao DW sobre atualizações, entretanto não possuem capacidade para informar quais foram exatamente os dados e fontes que foram alterados, podendo acarretar inconsistência nas Visões Materializadas. Contudo para amenizar esse problema, existem na literatura alguns algoritmos para manutenção incremental das Visões, nos ambientes de DW (Agner, 2000).

No estudo de Agner, 2000, foram levantados algoritmos da família ECA (*Eager Compensating Algorithm*), *ECA-Key e ECA-Local*, e ainda algoritmos da família *Strobe*, compostos pelo próprio *Strobe, T-Strobe, G-Strobe, e C-Strobe* (Agner, 2000).

A família de algoritmos ECA analisa basicamente as consultas realizadas do DW a apenas um sistema fonte. Seu código analisa um conjunto de requisições do DW a fonte de dados que ainda não obteve resposta. Se este agrupamento não estiver vazio, assim que o DW é informado de que houve atualização dos dados na fonte, o ECA inicia nova consulta para as requisições sem resposta. O algoritmo armazenará os dados coletados temporariamente, e fará a atualização das Visões Materializadas apenas quando o conjunto de requisições não respondidas estiver vazio novamente (Agner, 2000).

Algoritmos *Strobe*, da mesma forma como os ECA, verificam o conjunto de requisições não respondidas, porém ao invés de apenas uma fonte de dados, o *Strobe* permite integração a mais de um sistema fonte. A ação da ferramenta se dá assim que o DW recebe informação sobre atualização da fonte por inserção ou exclusão. Quando há exclusão de informação, o algoritmo analisa valores dos atributos chaves e inicia

atividade para remoção diretamente das mesmas informações na visão materializada. Quando é feita inclusão de novas informações, o *Strobe* deverá antes de iniciar a alteração, verificar qual foi a fonte de dados alterada. Para isto são geradas consultas e enviadas às fontes de dados relacionadas (Agner, 2000).

Para a manutenção do tipo *SWEEP*, trata-se de uma extensão dos algoritmos apresentados acima. Este se difere no aspecto que, diferente do ECA e *Strobe*, realiza um processamento de requisições apenas para atualizações que interferem diretamente no resultado de consultas, e realizam a compensação diretamente no DW. Tal processo é definido como *on-line error correction* e é aplicada removendo os efeitos das atualizações concorrentes quando são detectadas no DW (Agner, 2000).

4. Avaliação da aplicação de processos de manutenção

O *Data Warehouse* possui grande importância no processo decisório das organizações. Tal processo depende da qualidade dos dados fornecidos por fontes heterogêneas que são armazenadas em Visões materializadas permitindo agilidade nas operações de consulta. Para garantir a confiabilidade dessas informações, torna-se necessário a manutenção dessas Visões Materializadas (Agner, 2004). Todavia, durante o processo de modelagem dos DW, geralmente as dimensões são projetadas sem um planejamento adequado para receber alterações ao longo do tempo, podendo ocasionar problemas quando, por exemplo, um cliente precisa alterar o estado civil. Hokama (2004) cita que os envolvidos no desenvolvimento do projeto devem determinar como tais alterações devem ser gerenciadas pelo sistema, e quais as melhores alternativas para tal (Hokama, 2004).

4.1 Técnicas para manutenção incremental

As Visões Materializadas oferecem um acesso rápido as informações agilizando as consultas, entretanto as Visões Materializadas ficam desatualizadas quando os sistema fonte de dados são atualizados. Na grande maioria dos casos de atualização, apenas parte dos dados é alterado, e nestes casos pode ser considerado um desperdício de recurso recomputar a visão do zero (Gupta, 1995).
São descritos na subseção 4.1.1, estudos de casos que listam e identificam pontos positivos e negativos nos processos de manutenção incremental

4.1.1 Algoritmo ECA

Zhuge em seu estudo levanta anomalias que podem ser causadas após uma atualização de dados no sistema fonte em um pequeno espaço de tempo após o DW receber uma notificação sobre atualização dos sistema fonte, contudo antes de serem processadas na integra. São propostas técnicas de manutenção que podem ser aplicadas para impedir que as informações sejam afetadas e percam confiabilidade (Zhuge, 1995).

O ECA tem por princípio permitir que os sistema fonte de informação e as visões materializadas sejam atualizados simultaneamente. O algoritmo tem por função certificar que as visões materializadas no DW sempre estejam corretas. A verificação é realizada através de consultas de compensação enviadas com queries do DW após uma alteração no sistema de origem das informações, com o intuito de verificar se foram realizadas atualizações nas fontes de dados, após o envio da notificação para o DW,

contudo antes que essa operação tenha sido processada na Visões Materializadas (Ciferri, 2002). A utilização de queries para compensação aplica a atualização das Visões apenas após o recebimento da última consulta. Zhuge afirma que caso o processo de atualização das Visões seja feito após cada recebimento essa visão poderá tornar-se inválida, tornando então o algoritmo inconsistente (Zhuge, 1995).

Ainda em seu estudo, Zhuge (1995) afirma que apesar de ser muito consistente, o ECA não é completo. Durante o recebimento das respostas das requisições enviadas, algumas modificações podem passar despercebidas pelo algoritmo. No entanto o ECA conta com variações dos algoritmos que apresentam algumas diferenças no funcionamento (Zhuge, 1995).

4.1.1.1 ECA-KEY

O ECA-KEY possui algumas melhorias em comparação ao ECA. Agner levanta que o algoritmo citado introduz o conceito de atualizações locais. As exclusões são processadas diretamente no DW sem necessidade do envio de consulta para as fontes, e as inserções são processadas sem que seja preciso enviar as queries de compensação como ocorre no ECA (Agner, 2000).

Segundo Zhuge, o DW ainda precisará enviar consultas as fontes, e dessa forma as anomalias poderão ocorrer. No entanto essas anomalias nos dados geram tuplas duplicadas ou ausentes (Zhuge, 1995). Contudo as tuplas ausentes seriam excluídas futuramente em um processamento de atualizações concorrentes, enquanto as tuplas duplicadas seriam identificadas e não incluídas na Visões Materializadas. Por este motivo não se faz necessário a utilização das consultas compensadoras incorporadas as queries enviadas para a fonte de dados (Agner, 2000).

4.1.1.2 ECA-Local

O algoritmo *ECA-Local* faz possui características similares aos outros algoritmos ECA. O ECA evita a anomalia que pode ocorrer nos dados quando são enviadas consultas para a fonte, utilizando queries de compensação. Introduzindo o conceito de atualizações locais, no caso de exclusões, o *ECA-Key* não faz uso dessas consultas de compensação pois não são enviadas consultas as fontes (Zhuge, 1995).

O *ECA-Local* faz a utilização das compensações do ECA combinadas com as atualizações locais, produzindo um algoritmo que pode ser utilizado em vários cenários.

4.1.2 Avaliação do estudo ECA

Para avaliar o algoritmo Zhuge propõe em seu estudo uma análise da performance do algoritmo ECA. Em seu trabalho foram levantadas formas de manter forte consistência das Visões no DW, e o algoritmo de manutenção incremental, mais especificamente, o ECA é uma delas (Zhuge, 1995).

Durante o desenvolvimento do estudo, Zhuge propõe um comparativo do ECA com a manutenção por Recomputação da Visão (RV) onde será avaliado quais cenários são ideais para aplicação de cada caso (Zhuge, 1995).

Para o estudo foram mensurados os fatores:

- Quantidade de mensagens enviadas entre a fonte de dados e o DW (M)
- Total de bytes enviados da fonte para o DW (B)

- Número de operações de I/O realizadas (*IO*)
- Total de atualizações no sistema fonte de dados (k)

Baseando-se pelo total de número de mensagens enviadas, foram tomadas por exemplo uma quantidade *k* de atualizações na fonte, considerando que o DW envia mensagens a fonte a cada *s* novas alterações, seriam no mínimo duas queries podendo chegar até o dobro de atualizações *k*. No entanto, o ECA sempre enviará mensagens para cada nova atualização *k*, tornando a situação menos favorável para o algoritmo que, independente do cenário teria *s* como o dobro de *k*, ou seja, o número de mensagens seria igual ao dobro do número de atualizações (Zhuge, 1995).

Ao realizar a análise pela quantidade de bytes transferidos e posteriormente por total de operação *input* / *output* foi definido um cenário mais específico composto de três updates, U_1, U_2, U_3. Para aplicar RV, o melhor cenário é dado quando há Recomputação apenas quando U_3 estiver finalizado, enquanto o pior cenário se dá quando após cada update houver Recomputação. No caso do ECA, quando as atualizações são realizadas com espaço de tempo, não há necessidade de enviar as mensagens de compensação, o que se tornaria a o melhor cenário. Entretanto em um cenário mais amplo proposto no estudo, Zhuge se baseia nas análises realizadas para informar que em termos de análise da quantidade de bytes transferidos, o ECA se mostra muito eficiente em casos onde os relacionamentos envolvem mais de 5 tuplas.

Em um novo teste realizado, a cardinalidade dos relacionamentos foi elevada para 100. Neste ponto observou-se que na pior situação para o ECA, onde os dados seriam atualizados antes que as respostas chegassem de volta ao DW, seria necessário compensar cada uma das atualizações realizadas, podendo aumentar ao quadrado a quantidade de dados transmitidos ao número de atualizações, enquanto RV precisaria apenas de 30 ou mais *updates*. Neste processo foi observado que independentemente da quantidade de atualizações realizadas na fonte, caso haja apenas uma vez, devido a frequência de atualização adotada. Quanto menor o tempo entre as atualizações, maior seria o custo (Zhuge, 1995).

Por fim, foi avaliado a quantidade de operações *IO* realizado por cada uma das técnicas de manutenção, identificando que o ECA apresenta grande vantagem sobre RV quando as relações são grandes. O RV apresentou vantagem nas relações inferiores a três blocos de memória.

Ao final do estudo, Zhuge conclui que Recomputar a Visão para novas alterações apresentaria vantagem apenas para pequenas bases, com poucos relacionamentos. No entanto para cenários maiores, o ECA apresenta vantagem com relação a quantidade de mensagens enviadas entre fonte e DW, total de trafego de dados, e operações de *IO*.

4.2.1 Algoritmo Transaction-SVM

No contexto de manutenções incrementais, foram desenvolvidos diversos algoritmos de apoio a técnica (Zhuge, 1995).

Em seu estudo, Agner (2004) propõe a utilização do algoritmo T-SVM (*Transaction Algorithm for Scheduling Warehouse View Maintenance*), este adaptado do algoritmo SVM, projetado para utilização para proporcionar forte consistência em cenários de DW com diversas fontes de dados.

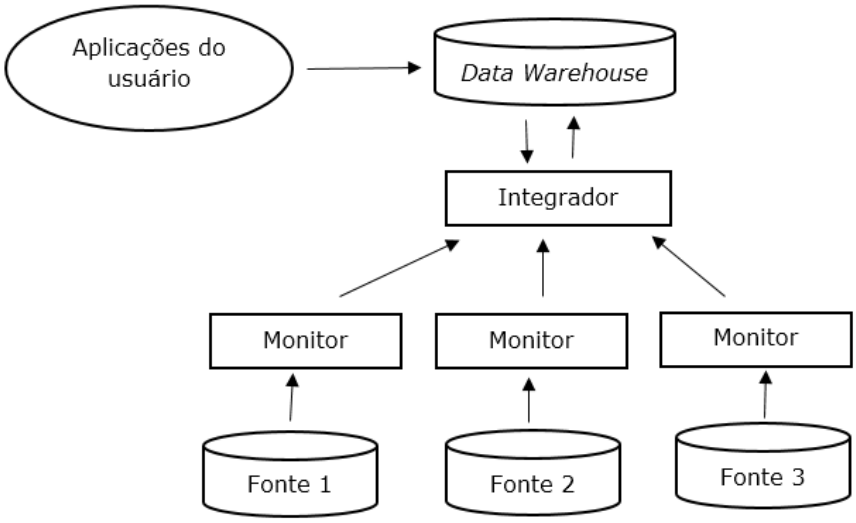

Figura 2. Arquitetura de um Data Warehouse
Fonte: Adaptado de Agner (2004).

Para simplificar o estudo do código, Agner propõe o algoritmo T-SVM que contempla monitores nas fontes de dados e no DW (Agner, 2004).

Os monitores implementados têm por função simplificar o método de comunicação e atualização do DW. O monitor vinculado as fontes de dados são responsáveis por unificar todas as alterações realizadas e enviá-las de forma única para o DW, além de receber consultas que o DW faz a fonte, e retorná-las com o resultado processado. Em se tratando de melhorias, o monitor integrado ao DW analisa toda notificação de atualização recebida das fontes. Tal função é desempenhada para otimizar a manutenção a ser realizada. Suponhamos que uma determinada tupla será atualizada e posteriormente excluída. Sem a ação do monitor, seriam enviadas as mensagens para atualização e exclusão dessas informações. Por este motivo é realizada a remoção de todos os pares de inserção e exclusão de uma mesma tupla definidas para serem processadas na ordem de inserção e exclusão (Agner, 2004).

Como forma de prevenir anomalias de dados, os monitores do DW desempenham uma rotina de análise no sistema fonte. Agner cita que sempre que as fontes estão processando uma requisição enviada pelo DW, existe a possibilidade concorrente de uma nova inserção de dados. O tratamento encontrado para minimizar os efeitos dessa deformidade nos dados é processado assim que o monitor do DW recebe a resposta da pesquisa realizada pelo mesmo a fonte. O monitor verifica se durante o tempo em que a requisição estava sendo processada houveram atualizações. Esta consulta é realizada através de um conjunto que armazena as atualizações na fonte que ainda não foram atualizadas. A compensação é realizada, caso necessário, no próprio DW, sem necessidade de enviar novas consultas as fontes relacionadas (Agner, 2004).

Por fim, ainda quando o monitor do DW recebe mensagem de exclusão, tem por procedimento informar a ação em uma lista de ações, contudo quando há um alerta de inclusão o monitor consulta as fontes dos dados relacionados (Agner, 2004).

4.2.2 Avaliação do estudo T-SVM

Em seu estudo, Agner (2004) propõe análise do algoritmo T-SVM em comparação com demais algoritmos, bem como ECA, *Strobe,* e *SWEEP*. Foram avaliados os algoritmos de manutenção incremental devido as vantagens apresentadas quando se comparadas as manutenções por Recomputação e Replicação. Na Recomputação, Agner conclui que demanda um tempo maior para realização do processo, enquanto a técnica de Replicação consome grande espaço para armazenamento (Agner, 2004).

Agner levanta que em comparação aos demais algoritmos estudados, o SVM e o T-SVM sabe exatamente quando devem atualizar as Visões. Os algoritmos da família *Strobe* e *SWEEP* só atualizam as Visões quando não estão sendo realizadas consultas concorrentes, nos permitindo afirmar que dependem do volume de informações de atualização recebidas para poder incorporá-las nas Visões. Por sua vez, os algoritmos SVM e T-SVM executam atualizações em períodos de tempo programados, mesmo que no momento estejam sendo processadas consultas, sem necessidade de interrompe-las (Agner, 2004).

O algoritmo SVM, T-SVM e família *Strobe* apresentam restrição por conta do tipo de atualização realizada. Deve constar junto a definição da visão, os atributos chave para cada relação base do DW, contudo devido a essa necessidade o número de mensagens para processamento na visão pode ser reduzido devido ao fato de que as exclusões podem ser realizadas diretamente na Visão Materializada do DW, sem necessidade de envio de consulta para as fontes, otimizando o processo de atualização (Agner, 2004).

5. Considerações finais

A partir dos estudos realizados neste trabalho, verifica-se a relevância tecnológica do uso do DW como ferramenta essencial para tomada de decisões gerenciais em organizações, e devido a esse propósito a necessidade de manter os dados confiáveis e sempre atualizados cresce na mesma proporção.

Com base nos estudos de caso e técnicas de manutenção apresentadas, pode-se observar que as técnicas de manutenção incremental podem oferecer níveis de consistência forte sem a necessidade de gerar replicações ou reprocessamento de todos os dados da Visão. Os algoritmos propostos passam por melhorias com o intuito de aprimorar cada vez mais a manutenção. O T-SVM estudado é parte do algoritmo SVM para promover forte consistência de dados.

Dentre os algoritmos levantados, para tratar possíveis anomalias dos dados, o ECA envia consultas de compensação as fontes sempre que há uma atualização. Contudo o T-SVM, apesar de realizar a manutenção apenas em períodos programados, unifica todas as mensagens a serem enviadas as fontes, realiza a compensação e exclusões localmente no DW o que impacta em uma redução do número de consultas a serem processadas.

Ao longo do desenvolvimento deste trabalho, verificou-se a existência de uma grande demanda por informações gerenciais de qualidade, que é suprida por avanços das técnicas e melhorias dos algoritmos para manutenção incremental, visando a confiabilidade dos dados e agilidade na consulta por informações gerenciais no DW. Contudo poucos estudos que envolvem o tema DW citam sobre técnicas de manutenção, e nestes não foi possível localizar informações sobre custos de implementação da

solução no DW. Em diversos estudos de caso voltados a implantação de DW não são especificadas as técnicas de manutenção no projeto, não deixando claro sequer sobre sua utilização.

O que pode-se sugerir para trabalhos futuros seria estudo para implementação de técnicas de manutenção em DW reais, com o intuito de realizar levantamentos de custo, tempo de desenvolvimento, métricas voltadas ao desempenho das técnicas de manutenção com relação a quantidade de mensagens trocadas entre fonte de dados e DW, análise da confiabilidade das informações geradas em consultas gerenciais estudados ao longo de um determinado período de tempo, a fim de determinar quais seriam as soluções que apresentam melhor custo benefício, e quais seriam as vantagens e desvantagens de utilizá-las no projeto.

Referências:

Agner, L. T. W "Manutenção Incremental de Visões Materializadas em Ambientes Data Warehouse" Curitiba 2000

Agner, L. T. W "T-SVM: Uma abordagem para manutenção de visões materializadas em ambientes *data warehouse*" Guarapuava 2004

Bischoff, J. Alexander, T. "Data Warehouse: Pratical Advice from the Experts" 1997

Chaudhuri, S. Dayal, U. "An Overview of Data Warehousing and OLAP Technology" Acm Sigmod Record, Volume 26, páginas 65 a 74 1997

Carvalho, G. T "Aplicações de práticas ágeis na construção de data warehouse evolutivo" Instituto de Matemática e Estatística da Universidade de São Paulo 2009

Ciferri, C. D. A. "Distribuição dos Dados em Ambientes de Data Warehousing: O Sistema WebD2W e Algoritmos Voltados à Fragmentação Horizontal dos Dados" Recife 2002

Dill, S. L."Uma Metodologia Para Desenvolvimento De Data Warehouse E Estudo De Caso"https://repositorio.ufsc.br/xmlui/handle/123456789/82897 acesso em 07/06/2017 19:30 2002

Ferreira, R. G. C"DataWarehouse na Prática: Fundamentos e Implantação**"** Universidade Federal do Rio Grande do Sul Instituto de Informática 2002

Fortulan, M. R, Filho, E. V. G "Uma proposta de aplicação de business intelligence no chão-de-fabrica" Gest. Prod. vol. 12 no.1 São Carlos 2005

Gupta, A Mumick, I. S. "Maintenance of Materialized Views: Problems, Techniques, and Applications" 1995

Gupta, P. Mata-Toledo R. A. Monger D. Morgan"Database Development Life Cycle". ftp://ftp.repec.org/opt/ReDIF/RePEc/rau/jisomg/SP11/JISOM-SP11-A1.pdf Acesso em: 07/06/2017 21:30

Hokama, D. D. B Camargo, D. Fujita, F Fogliene, J. L. V “A modelagem de dados no ambiente Data Warehouse” Universidade Presbiteriana Mackenzie, Faculdade de Computação e Informatica 2004

IBM Knowledge Center “https://www.ibm.com/support/knowledgecenter/en/SSULQD_7.2.0/com.ibm.nz.dbu.doc/c_dbuser_materialized_views.html” acesso em 28/10/2017 15:00

Javed, A; Rafique, S. S. “Data Warehouse Maintenance. Improving Data Warehouse Performance through Efficient Maintenance”. Lulea University of Technology 2006

Kimball, R. Ross, M. “The Data Warehouse Toolkit The Definitive Guide to Dimensional Modeling” Indianapolis 2013

MattiodaI, R. A Favaretto, F. “Qualidade da informação em duas empresas que utilizam Data Warehouse na perspectiva do consumidor de informação – um estudo de caso”http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0104-530X2009000400013&lng=en&nrm=iso acesso em 04/06/2017 21:06 2009

Miranda, W. “Diferença entre VIEWS e MATERIALIZAD VIEWS no Oracle” aprendasql.com 2014

Paim, F. R. S“Uma metodologia para definição de requisitos em sistemas Data Warehouse”Universidade Federal de Pernambuco 2016

Saccol, D. B. “Materializações de Visões XML” Universidade Federal do Rio Grande do Sul 2001

Santos, R. S Almeida, A. L Tachinardi, U Gutierrez, M. A “Data Warehouse para a Saúde Pública: Estudo de Caso SES-SP” X Congresso Brasileiro de Informática em Saúde 2006

Schwaickardt, E. “O fator manutenção no ciclo de vida de data warehouse” SlideShare 2013

Zhuge, Y. Garcia-Molina, H. Hammer, J. Widom, J. “View Maintenance in a Warehousing Environment.” Standford, CA 94305-2140, USA 1995