9ª MOSTRA ACADÊMICA UNIMEP
08 a 10/11 de 2011
Tema: "Ambiente e Sustentabilidade"

9º Congresso de Pós-Graduação

ANÁLISE E OTIMIZAÇÃO NO ALGORITMO SVM

Autor(es)

RUBENS THIAGO DE OLIVEIRA

Co-Autor(es)

MARINA TERESA PIRES VIEIRA

Orientador(es)

MARINA VIEIRA

1. Introdução

O sucesso das organizações está diretamente relacionado aos conhecimentos estratégicos e táticos do negócio. Os sistemas de apoio a decisão tornaram-se rapidamente um ponto chave para obter vantagem competitiva (GUPTA; MUMICK, 2005). Analistas de negócios utilizam esses sistemas para detectar tendências de mercado. Normalmente, os sistemas de apoio a decisão utilizam informações históricas e, portanto, as bases de dados tendem a ser volumosas ao longo do tempo. Para a utilização dos recursos adequados à tomada de decisão utiliza-se um data warehouse. Segundo Inmon (2005) o data warehouse é uma coleção de dados orientados por assuntos, integrados, variáveis conforme o tempo e não voláteis, para dar suporte ao processo de tomada de decisão; trata-se de um processo com ciclo de vida contínuo que aglutina dados de fontes heterogêneas, incluindo dados históricos. Numa arquitetura simplificada de um DW, as informações são carregadas periodicamente a partir de múltiplas bases de dados transacionais. Os dados relevantes são sumarizados, agrupados e armazenados no DW, para serem utilizados pelos usuários para a tomada de decisão. As consultas dos sistemas de apoio a decisão muitas vezes possuem instruções complexas sobre as fontes de informação. O tamanho e a complexidade das consultas podem provocar um grande consumo de tempo para que sejam concluídas, o que muitas vezes pode inviabilizar a tomada de decisão (ASHADEVI; BALASUBRAMANIAN, 2008). Um requisito básico para o sucesso de um DW é a capacidade de fornecer aos analistas de negócio, informações precisas e consolidadas, com tempos de resposta eficientes. Um dos mecanismos adotados para atender essa demanda é a utilização de visões materializadas (KARDE; THAKARE, 2010a). A visão materializada é uma consulta cujo resultado já está computado e armazenado na base de dados (GUPTA; MUMICK, 1999). As consultas que puderem utilizar as visões já armazenadas podem ser executadas de forma muito mais rápida, sendo que, para consultas complexas envolvendo grandes volumes de dados, esta alternativa favorece dramaticamente os resultados: de horas ou dias para segundos ou minutos conforme citam Zhang, Yao e Yang (2001). As visões materializadas, apesar de melhorarem o desempenho de consultas ao DW, podem envolver altos custos na sua manutenção e atualização, o que pode prejudicar o desempenho total do DW (ZHANG; YAO ;YANG, 2001). Além disso, depara-se com a questão da utilização de espaço em disco adicional para armazenamento das visões materializadas, que representa um aumento de consumo de recursos do DW (ASHADEVI; BALASUBRAMANIAN, 2008). O uso de visões materializadas nos data warehouses requer análise de alguns aspectos: ? A seleção das visões mais adequadas a serem materializadas, levando em conta uma análise dos custos de manutenção e os benefícios de cada visão e os algoritmos disponíveis; ? A utilização das visões materializadas para responder às consultas; ? Mecanismos que permitem a propagação correta quando da atualização das fontes de dados, para vários tipos de visões; ? A manutenção das visões de forma dinâmica e incremental; ? A criação de algoritmos que permitam a manutenção das visões de maneira autônoma. Na figura 1, são mostradas as várias políticas adotadas para a seleção de visões a serem materializadas, conforme citado em Li (2010). Configuração

evolutiva: Política que permite ajustar a seleção de visões a serem materializadas, baseada na semântica de cache. Pontos significativos nessa abordagem: são a política do cache e a política de reutilização. Na política de cache, as consultas são informadas pelo analista e o algoritmo determina se as consultas informadas podem ser utilizadas para atender uma nova visão a ser materializada ou se existe um fragmento de uma consulta disponível em disco, onde o custo de execução seja menor. Por outro lado, a política de reutilização descreve como as visões podem ser materializadas para acelerar os tempos de respostas, (KOTIDIS; ROUSSOPOULOS, 2001), apud (LI, 2010). Restrições de recursos: Devido ao grande volume de dados dos DWs, o espaço de armazena-mento é geralmente a primeira questão a ser considerada. No entanto, em relação às visões materializadas, o limite de tempo na atualização das visões materializadas também se torna vital. A materialização também pode tornar o tempo de manutenção e atualização longo e, portanto, diminuir a disponibilidade do DW (ZHANG; YAO; YANG, 2001). Seleção de inter-relacionamento: A política de seleção de visões a materializar utilizando o inter-relacionamento, se baseia na contenção de consultas, pois uma visão selecionada à materialização pode tornar uma visão já materializada sem utilidade, devido a sobreposições de informações. A política verifica também a dependência entre as agregações de um cubo de dados, onde consultas cujas informações agregadas e armazenadas na base tenham dependência entre as visões e possam ser re-utilizadas na materialização. Para a identificação das dependências dos inter-relacionamentos das consultas, pode-se utilizar um diagrama lattice, que representa as relações de dependências entre as visões. As dependências também podem ser visualizadas utilizando ferramentas de gerenciamento de banco de dados. (LI, 2010). Metas de otimização: São as políticas de seleção das visões materializadas baseadas em tem-po de resposta. Muitas abordagens existentes optam por algumas outras medidas relevantes, como o espaço para armazenamento e tempo para atualização das visões materializadas (GUPTA; MUMICK, 2005) e (KARDE; THAKARE, 2010b). Como os custos financeiros para aquisição de armazenamento estão diminuindo e a janela de atualização das visões materializadas se encolhendo, muitas abordagens buscam melhorar os custos de atualização das visões materializadas ao invés da melhoria no tempo de resposta. Outra abordagem utilizada também é incorporar ambos, os custos de avaliação do tempo de resposta e os custos de atualização da visão materializada. Visões candidatas a materialização: Para a materialização de visões, existem alguns critérios que podem ser adotados para restringir as consultas a serem materializadas. A necessidade da criação de um índice para melhorar o desempenho no tempo de resposta da consulta pode ser um dos critérios adotados nessa política. Outro critério utilizado, baseia-se num conjunto de consultas que possuam sub-expressões em comum e que possam ser reutilizadas na materialização. Para a identificação e modelagem das sub-expressões comuns, pode utilizar-se o diagrama de lattice. Existem também critérios que baseiam-se no histórico de uso de determinada consulta, como citado em Neves (2009). A combinação entre políticas pode ser utilizada durante o desenvolvimento de um algoritmo, visando atender a critérios específicos do mesmo, vale citar como exemplo Kotidis e Roussopoulos (2001) que utilizam em seu algoritmo a política de configuração evolutiva, modelagem de inter-relacionamento e metas de otimização.

2. Objetivos

Pretende-se, no trabalho proposto, refinar a estratégia adotada por Neves (2009) para a seleção de visões a materializar, na fase de projeto de data warehouses, ou mesmo propor uma nova estratégia, visando obter melhorias nos custos de execução de consultas em data warehouses usando as visões materializadas. Para alcançar esses objetivos estão sendo estudados os principais algoritmos de seleção de visões materializadas da literatura e em particular o algoritmo de Neves (2009), para poder identificar as principais características dessas abordagens. A escolha pela estratégia de Neves (2009) deve-se ao fato de ser uma abordagem simples e que contempla aspectos importantes para escolha de visões a materializar. Além disso, é uma das poucas abordagens que enfoca materialização de visões empregada na fase de projeto do DW, que é também o enfoque deste trabalho.

3. Desenvolvimento

Para subsidiar o desenvolvimento do trabalho está sendo realizada uma revisão da literatura, focando, principalmente, as políticas adotadas por vários autores para a seleção de visões a materializar para data warehouses. As abordagens de seleção de visões para data warehouses são, na sua maioria, voltadas para DWs que estão em fase de uso. Um diferencial da abordagem de Neves (2009) é que ela foi proposta para uso na fase de projeto do DW, antes mesmo de sua implantação. Neves propõe que, juntamente com o projeto do DW, sejam definidas as visões para auxiliar a execução de consultas no DW na sua fase inicial de uso, antes que se tenha um histórico de consultas efetivamente realizadas no DW que possam subsidiar a seleção das visões a materializar. A motivação para o desenvolvimento de uma nova proposta de seleção de visões para materializar, em data warehouses, surge da necessidade de dar continuidade aos trabalhos iniciados por Neves (2009), a fim de refinar sua estratégia, buscando avaliar a viabilidade de incorporar outros elementos importantes existentes em outras abordagens da literatura. Pretende-se, portanto, focar a técnica não somente no desempenho das consultas, mas também em outros aspectos, tais como nos custos de manutenção das visões. O estudo do uso de visões materializadas para acelerar o processamento de consultas em DWs se estende a mais de 20 anos e os principais sistemas de banco de dados comerciais (DB2, Oracle, SQL Server) suportam visões materializadas. No entanto, os custos de armazenamento podem ser elevados e a atualização das informações nas visões materializadas podem se tornar onerosas ao DW (ZHOU et al., 2007). O problema para selecionar visões a serem materializadas para responder a consultas para melhorar o desempenho do DW, tem sido tradicionalmente estudado sob o nome de seleção de visões. Kotidis e Roussopoulos (2001)

apresentam um sistema aplicado ao gerenciamento dinâmico de visões. O sistema, denominado DynaMat, determina se as consultas informadas pelo analista podem ser utilizadas para atender uma nova visão a ser materializadas ou se existe um fragmento de uma consulta disponível em disco, onde o custo de execução seja menor. Zhang, Yao e Yang (2001) propuseram uma abordagem completamente diferente, a utilização de um algoritmo genético, para escolher as visões a serem materializadas e demonstraram que é prático e eficaz em comparação com as abordagens heurísticas. Gupta e Mumick (2005) desenvolveram um framework para a resolução do problema da seleção de visões num DW. A heurística apresentada, otimiza o tempo de resposta das consul-tas para casos especiais de cenários de DW, onde a formulação do problema é dado através de expressões AND-DAG e ANDOR-DAG. Yang e Chung (2006) desenvolveram um cluster baseado no algoritmo ASVMRT. Neste algoritmo as tabelas são reduzidas, calculadas e agrupadas utilizando técnicas de cluster e visões materializadas, os dados, são calculados com base nas tabelas reduzidas ao invés de tabelas originais. Como resultado os custos de execução e atualização são otimizados. Zhou et. al (2007) propõem uma estratégia de materialização mais flexível destinada a reduzir o espaço de armazenamento e os custos de manutenção. Na visão materializada é selecionado apenas um subconjunto de tuplas, as mais acessadas. O subconjunto de tuplas a serem materializadas podem ser alteradas dinamicamente, manualmente ou controladas automaticamente por um gerenciador interno de cache. Ashadevi e Balasubramanian (2009) desenvolveram um framework para a seleção de visões que explora as métricas de custo associadas às visões materializadas entre elas, a frequência de uso, os custos de execução, a atualização das tabelas de base, os custos de manutenção da visão materializada e as restrições de espaço de armazenamento. Antes de selecionar novas visões a serem materializadas o framework, remove as visões materializadas que apresentam pouca frequência de uso e com espaço de armazenamento alto. Depois o framework calcula o custo total da consulta, validando as métricas acima descritas e as consultas com o melhor custo são selecionadas para a materialização. Neves (2009) desenvolveu o algoritmo SVM, que seleciona visões a materializar com base na frequência de uso e no custo da materialização, durante o processo de criação do DW. O projetista do DW informa as consultas mais frequentes e o algoritmo analisa as políticas de freqüência de uso e custos na execução, selecionando as consultas que serão materializadas no DW. Ashadevi e Navaneetham (2010) apresentaram um framework para a seleção de visões que explora as métricas de custo associadas às materializações de visões, à freqüência das consultas, o custo dos acessos das consultas, à frequência de atualização das tabelas bases, o custo da atualização das visões e as limitações do espaço de armazenamento no DW. Karde e Thakare (2010b) propuseram dois algoritmos, um para a seleção e manutenção de visões a serem materializadas e o outro algoritmo é para a seleção de nós. Este último algoritmo determina em qual nó do ambiente distribuído a visão materializada deverá ser criada, atualizada ou para ser mantida. Vários parâmetros foram considerados: o custo da consulta, de manutenção, do tráfego de rede e limitações de espaço para armazenamento. Bhagat e Harle (2011) abordam a criação de dois algoritmos. O primeiro algoritmo é para a geração e manutenção da visão materializada, utilizando a abordagem de árvores. O segundo algoritmo é para a seleção de nó onde a visão materializada será distribuída e atualizada.

4. Resultado e Discussão

Para alcançar os resultados propostos neste trabalho foram realizados estudos do algoritmo SVM (NEVES, 2009). E também foi criado um laboratório para testes, utilizando a base de dados do Transaction Processing Performance Council - TPC-H (ONEIL, P.; ONEIL, B.; CHEN, 2009). A figura 2 apresenta a modelagem dimensional utilizada, sob o esquema estrela, utilizado nos testes. Para a geração do modelo físico da base de dados foi utilizado o gerenciador de banco de dados Oracle 11g. A base de dados TPC-H possui um conjunto pré-definido de consultas para testes, abaixo estão exemplificadas duas das consultas utilizadas no estudo. Q2.1: select sum(lo_revenue), d_year, p_brand1 from lineorder, zdate, part, supplier where lo_orderdate = d_datekey and lo_partkey = p_partkey and lo_suppkey = s_suppkey and p_category = 'MFGR#12' and s_region = 'AMERICA' group by d_year, p_brand1 order by d_year, p_brand1; Q3.1: select c_nation, s_nation, d_year, lo_revenue as revenue from customer, lineorder, supplier, zdate where lo_custkey = c_custkey and lo_suppkey = s_suppkey and lo_orderdate = d_datekey and c_region = 'ASIA' and s_region = 'ASIA' and d_year >= 1992 and d_year