

5 Conclusões e Trabalhos Futuros

Esta dissertação focou na sintonia automática de visões materializadas. O processo de criação automática deste trabalho é baseado no ciclo de observação, predição e reação. Esta dissertação apresentou heurísticas necessárias para criação de visão materializada que foram utilizadas em uma abordagem não-intrusiva ao código do banco de dados. Para tanto algumas medidas necessárias para criação das heurísticas de seleção de visão materializada foram definidas

Essas heurísticas foram implementadas na arquitetura para gerenciamento do projeto físico de banco de dados em Java, para o banco de dados SQL Server. Foram realizados testes para comprovar a eficiência dessas heurísticas. De fato, os testes mostraram que as visões materializadas criadas através das heurísticas na arquitetura proposta melhoraram o desempenho do sistema.

As principais dificuldades encontradas nesta dissertação dizem respeito a como calcular os fatores para avaliar o benefício de uma visão materializada. Como esta não existe fisicamente no banco de dados, foram propostas funções que estimam alguns custos, tais como o custo de criação da visão e o custo de uma varredura completa.

5.1. Principais contribuições

Uma das principais contribuições desta dissertação é a revisão do estado da arte na área de sintonia de banco de dados, em particular, na área de visões materializadas. Outra importante contribuição são as heurísticas para seleção de visões materializadas e a heurística para seleção final de visões. Essas heurísticas possibilitam a instanciação da abordagem não-intrusiva e *on-the-fly* para seleção e materialização de visões.

Para a heurística de seleção de visões materializadas foram expostas características que revelam como criar a visão, quais atributos das consultas devem ser usados e sobre quais consultas devem-se criar e materializar visões.

Para a heurística de benefício de visão materializada foram definidos e propostos fatores necessários para sua correta execução. Esses fatores foram

propostos nesta dissertação, pois não havia como calcular o benefício de uma visão que não existia fisicamente no banco de dados. A partir dessa heurística foi possível definir quais visões materializar.

Outra contribuição é a implementação dessas heurísticas na arquitetura utilizando a abordagem não-intrusiva para criação automática de visões materializadas. A arquitetura é implementada em linguagem Java e fornece suporte para o SGBD SQL Server 2008.

5.2. Trabalhos Futuros

Uma oportunidade para trabalho futuro é criar visões sobre consultas analisando os filtros das consultas. Conforme visto na seção 3.1.1, as visões são criadas tornando o campo de filtro da consulta em um campo de seleção na visão. Se o campo de filtro da consulta costuma usar um mesmo intervalo de valores, pode-se criar uma visão com um filtro utilizando esse intervalo. Para tanto seria necessário sempre manter duas medidas para definir o intervalo e verificar qual a frequência que as consultas utilizam esse intervalo.

É interessante rever o limite estipulado para materialização de visões. Pois, se a visão mv_3 do apêndice D fosse materializada na segunda, ela traria um benefício de 117.7262 (unidade de custo do otimizador) cada vez que a consulta 17 fosse executada, no caso, 16 vezes. Esse benefício é praticamente o mesmo benefício da visão mv_7, visão que foi materializada pela ferramenta dessa dissertação, utilizada pela consulta 1 que foi executada 14 vezes na mesma carga.

Outra oportunidade de trabalho futuro é avaliar o malefício de uma atualização nos dados da tabela base da visão. Toda vez que uma atualização é submetida, é necessário avaliar qual o custo adicional de atualizar a visão materializada atribuir esse custo ao benefício da visão. Desta forma podem-se avaliar quais visões já materializadas não estão sendo mais úteis, trazendo até um custo maior para a carga de trabalho, por sofrerem muitas atualizações.

Outra oportunidade de trabalho é fazer a reescrita de consulta automática para utilizar as visões materializadas para o SGBD PostgreSQL. Este SGBD não faz a reescrita automaticamente como SQL Server, Oracle e DB2. Sem a reescrita, as visões criadas por esta dissertação não trariam benefícios, pois não seriam usadas.