

8

Conclusões e Trabalhos Futuros

As aplicações de bancos de dados têm se tornado cada vez mais complexas e variadas. Estas podem ser caracterizadas pelo grande volume de dados e pela demanda elevada em relação ao tempo de resposta das consultas e à vazão (*throughput*) das transações. Neste contexto, a sintonia (*tuning*) do projeto físico de bancos de dados tem se revelado ainda mais importante, influenciando diretamente no desempenho dos sistemas de bancos de dados.

Contudo, realizar o ajuste do projeto físico de forma manual tem se tornado uma tarefa suficientemente complicada para as aplicações atuais. Esta tarefa requer um profundo conhecimento acerca dos detalhes de implementação dos sistemas de gerenciamento de bancos de dados (SGBDs), das características dos dados armazenados, das aplicações e da carga de trabalho (conjunto de consultas e atualizações) submetida ao SGBD (Bruno07a, Luh07).

Neste contexto, os principais fabricantes de SGBDs oferecem ferramentas para suportar a sintonia automática (ou auto-sintonia) do projeto físico de bancos de dados (e.g., (Zilio04, Bruno05)). A idéia é auxiliar os administradores (DBAs) por meio da análise automática da carga de trabalho e, com base nesta análise, recomendar a criação ou remoção de índices, a manutenção de visões materializadas e um eventual particionamento de grandes tabelas. Entretanto, tais ferramentas adotam uma abordagem estática (*offline*) e transferem para o DBA tarefas de sintonia importantes e decisões relativas a quando modificar o esquema e quais de fato são as modificações necessárias. Além disso, são soluções *ad-hoc*, voltadas para SGBDs específicos e que exigem participação de DBAs especialistas. Adicionalmente, os DBAs precisam determinar o momento em que o sistema necessita de ajustes, a fim de iniciar uma sessão de sintonia (executando explicitamente a ferramenta). Para isso, os DBAs necessitam monitorar continuamente o desempenho do sistema e, sempre que julgarem necessário, executar uma sessão de sintonia. Todavia, utilizar a ferramenta de sintonia e executar os ajustes de forma freqüente resulta em desperdício de recursos, podendo degradar o desempenho do sistema. Por outro lado, executar as tarefas de sintonia esporadicamente pode conduzir à perda de importantes oportunidades de melhora de desempenho (Sal04). Desta forma, em ambientes

dinâmicos, com consultas *ad-hoc*, por exemplo, torna-se bastante complicado identificar as estruturas de armazenamento e acesso a dados potencialmente úteis, mesmo com o auxílio das ferramentas de sintonia.

Recentemente, algumas iniciativas apresentaram descrições de protótipos que implementam certas funcionalidades de sintonia automática do projeto físico. Entretanto, estes trabalhos adotam uma abordagem intrusiva, funcionam apenas com um SGBD específico, requerem chamadas adicionais ao otimizador e consideram somente a utilização de índices secundários.

Neste trabalho, propomos uma abordagem não-intrusiva para o problema da manutenção automática e *online* do projeto físico de bancos de dados. Nas próximas seções, apresentaremos as principais contribuições desta tese e algumas oportunidades para trabalhos futuros .

8.1

Análise das Principais Contribuições

As principais contribuições desta tese são:

1. Uma nova taxonomia para a classificação das pesquisas em auto-sintonia de bancos de dados.
2. Uma abordagem não-intrusiva para a manutenção automática do projeto físico de bancos de dados relacionais.
3. Uma instanciação da abordagem proposta para solucionar o problema da manutenção automática das estruturas de índices.
4. Uma instanciação da abordagem proposta para solucionar o problema da “clusterização” alternativa de dados.
5. Uma arquitetura não-intrusiva para suportar a manutenção automática do projeto físico de bancos de dados.
6. Uma implementação da arquitetura proposta.

8.1.1

Uma Taxonomia para as Pesquisas em Auto-Sintonia de Bancos de Dados

Nesta tese, propomos e justificamos uma nova taxonomia para a classificação das pesquisas em auto-sintonia de bancos de dados. Esta nova classificação baseia-se em duas dimensões ou eixos: (1) a abrangência da solução (Global vs Local) - a dimensão X - e (2) o grau de independência da solução em relação ao código do SGBD (Não-Intrusiva vs Intrusiva) - a dimensão Y.

8.1.2

Uma Abordagem Não-Intrusiva para a Manutenção Automática do Projeto Físico de Bancos de Dados Relacionais

A abordagem para a manutenção das estruturas que compõem o projeto físico de bancos de dados relacionais, proposta nesta tese, possui as seguintes características:

1. Não-intrusiva: a abordagem proposta é completamente desacoplada do código do SGBD utilizado, não sendo necessárias alterações no código fonte do SGBD.
2. Genérica: a abordagem pode ser utilizada com qualquer SGBD.
3. Automática: a abordagem independe de interações com DBAs.
4. *On-The-Fly*: executa continuamente durante a operação normal do SGBD utilizado.

Abordagem proposta apresenta, ainda, as seguintes contribuições:

1. Define e utiliza os conceitos de plano hipotético e otimização hipotética.
2. Fornece um conjunto de algoritmos gulosos e baseados em programação dinâmica para seleção das estruturas de acesso que compõem uma configuração de projeto físico.
3. Define e utiliza (no processo de manutenção das estruturas de acesso) os conceitos de relevância e generalidade das estruturas de acesso. Relevância consiste no grau de importância de uma determinada expressão SQL em relação às demais. Na prática, algumas expressões são mais importantes do que outras. Já o conceito de generalidade representa a qualidade de uma estrutura de acesso para carga de trabalho como um todo.
4. Define e utiliza (no processo de manutenção das estruturas de acesso) o conceito de estabilidade de uma configuração. Configurações instáveis conduzem a freqüentes criações e remoções das mesmas estruturas de acesso. Logicamente, este fato gera uma sobrecarga desnecessária, podendo, até mesmo, levar à perda de desempenho (Luh07). A fim de produzir configurações estáveis, o conceito de época, inicialmente proposto em (Kai04, Luh07), foi estendido e utilizado.

8.1.3

Uma Instanciação da Abordagem Proposta para Solucionar o Problema da Manutenção Automática das Estruturas de Índices

A abordagem foi instanciada com a finalidade de possibilitar a manutenção automática das estruturas de índices. As estratégias adotadas nesta instanciação proporcionaram ainda as seguintes contribuições:

1. Leva em consideração as restrições de espaço físico disponível para a materialização das estruturas de índice.
2. Utiliza uma heurística integrada para a seleção e acompanhamento de índices (denominada Heurística *HISAI*). Esta heurística produz uma quantidade de índices candidatos substancialmente menor do que a heurística apresentada em (Loh00) e implementada em (Sal04, Sal05, Cos05, Morelli06a, Lif06, Luh07).
3. Cada consulta SQL é otimizada uma única vez.
4. Pode ser executada em uma máquina distinta daquela onde o SGBD é hospedado.
5. Acompanha o nível de fragmentação das estruturas de índices reais.
6. Considera a atribuição proporcional de benefícios. Assim, cada índice recebe como benefício apenas o valor referente à sua real participação na otimização de uma determinada consulta.
7. Considera a pré-criação de índices envolvidos em chaves primárias e estrangeiras como forma de melhorar a qualidade do processo de seleção *on-the-fly* de índices e acelerar a obtenção de uma configuração de índices estável.

8.1.4

Uma Instanciação da Abordagem Proposta para Solucionar o Problema da Manutenção Automática de *Clusters* de Dados Alternativos

A abordagem foi instanciada com a finalidade de possibilitar a manutenção automática de *clusters* alternativos de dados. As estratégias adotadas nesta instanciação proporcionaram ainda as seguintes contribuições:

1. Utiliza visões materializadas para construir os *clusters* de dados.
2. Possibilita a duplicação de dados de forma total ou parcial.

3. Leva em consideração as restrições de espaço físico disponível para a materialização dos *clusters* de dados.
4. Pode ser executada em uma máquina distinta daquela onde o SGBD é hospedado.

8.1.5

Uma Arquitetura Não-Intrusiva para a Manutenção Automática do Projeto Físico de Bancos de Dados

A arquitetura concebida realiza a manutenção automática do projeto físico de bancos de dados mediante a colaboração entre os agentes de *software*, os quais utilizam *drivers* para obter e manipular informações do SGBD utilizado.

A seguir, enumeramos as principais características da arquitetura proposta:

1. Não-intrusiva: a arquitetura não requer alterações no código fonte do SGBD utilizado.
2. Genérica: a arquitetura pode ser utilizada com qualquer SGBD (desde que existam *drivers* instanciados para o SGBD que se deseja utilizar).
3. Automática: o funcionamento da arquitetura independe de interações com seres humanos.
4. *On-The-Fly*: executa continuamente durante a operação normal do SGBD utilizado.

As principais funcionalidades da arquitetura são:

1. Permite a manutenção (seleção, criação, remoção e reorganização) automáticas de estruturas de índices, visões materializadas e particionamentos de tabelas.
2. Possibilita a manutenção automática de *clusters* de dados alternativos.
3. Provê suporte para a manutenção pró-ativa de índices para consultas pesadas.
4. Permite guiar o DBA na reescrita de consultas.

8.1.6

Uma Implementação da Arquitetura Proposta

Um protótipo que instancia a arquitetura proposta para os SGBDs PostgreSQL 8.2, Oracle 10g e SQL Server 2005 foi implementado inteiramente em linguagem Java. Detalhes deste protótipo podem ser encontrados em (Monteiro08a).

8.2

Oportunidades para Trabalhos Futuros

A seguir descrevemos algumas oportunidades para trabalhos futuros:

1. Instanciar a abordagem proposta para realizar a manutenção automática de outras estruturas de acesso, tais como:
 - **Visões Materializadas.** Conforme discutimos na Seção 2.4, uma visão materializada pode ser vista como um *cache* (área de armazenamento temporário), ou como uma cópia dos dados que pode ser acessada rapidamente. Desta forma, uma visão materializada oferece acesso rápido aos dados, sendo que esta velocidade pode ser crítica em aplicações onde a quantidade de consultas é alta e a complexidade das visões elevada. Por outro lado, as alterações realizadas sobre os dados das tabelas base, sobre as quais uma visão materializada é definida, torna a visão desatualizada. Para que a visão possa estar novamente sincronizada com os dados das tabelas base, será necessário recriar a visão a partir dos dados da origem ou então, atualizá-la de forma incremental (Quass96). Contudo, este processo de atualização (ou manutenção) das visões materializadas consome tempo e recursos computacionais, podendo inclusive degradar o desempenho do sistema. Assim, a escolha do conjunto de visões materializadas a serem utilizadas deve buscar aliar um bom desempenho com um baixo custo de manutenção. Todavia, este também é um problema NP-Difícil (Agra00). Logo, a manutenção de um conjunto de visões materializadas que seja sempre adequado é um dos principais problemas envolvidos no projeto físico de bancos de dados (Agra00, Monteiro06b).
 - **Partições de Tabelas.** O particionamento de dados, conforme discutido na Seção 2.5, é um método que consiste em dividir fisicamente as grandes tabelas em diversos segmentos menores de dados, tornando o acesso aos dados mais rápido e seu gerenciamento mais fácil. Assim, o particionamento de grandes tabelas é uma das

mais importantes atividades relacionadas ao projeto físico de bancos de dados. Contudo, a manutenção automática do particionamento de grandes tabelas ainda é um problema pouco estudado.

2. Conceber uma estratégia para a manutenção conjunta de diferentes estruturas de acesso. O projeto e a definição de funções de utilidade que levem em consideração múltiplos critérios tem se revelado um grande problema. Além disso, a concepção de estratégias eficientes para combinar (integrar) diferentes componentes de sintonia autônomos (e locais) com a finalidade de obter comportamento global adequado continua sendo um problema em aberto.
3. Investigar a manutenção pró-ativa de índices e outras estruturas de acesso para consultas “muito pesadas”. Determinadas consultas, como consultas em aplicações OLAP, por exemplo, apresentam um tempo de execução bastante elevado, ou seja, são consultas bastante demoradas (“pesadas” ou de custo elevado). Tais consultas são de tal maneira demoradas que a criação de estruturas de índices adequadas trariam ganhos de desempenho mesmo que a consulta fosse executada uma única vez, e que o índice fosse excluído logo após sua execução. Em outras palavras, o benefício de se materializar a estrutura de índices é maior do que seu custo de criação em apenas uma execução da consulta. Neste contexto, seria interessante ter um histórico das execuções das consultas que possuem esse perfil e, com base nesse histórico, prever o momento em que uma dessas consultas seria executada novamente, com o objetivo de se antecipar a este evento e criar, de forma automática e pró-ativa, as estruturas de índices adequadas para acelerar a execução da referida consulta. Após a execução da consulta, o índice seria automaticamente removido. Dentre as técnicas que poderiam ser investigadas para se prever o momento em que uma consulta “pesada” seria executada, podemos destacar as Redes Neurais Artificiais, Regressão Linear e Cadeias de Markov.
4. Conceber e implementar um *Wizard* que seja capaz de orientar o DBA na reescrita de consultas. Uma vez que as consultas capturadas são armazenadas em uma metabase local, estas consultas poderiam ser analisadas e, a partir de uma base de regras, a ferramenta poderia sugerir ao DBA (por meio de alertas ou relatórios) oportunidades de reescrita que possam trazer ganhos de desempenho.
5. Investigar estratégias para o agendamento das tarefas de sintonia. As ações de criação e reconstrução das estruturas de acesso consomem re-

curso do SGBD e, caso sejam executadas em momentos em que o nível de atividades do SGBD esteja alto, podem até mesmo comprometer o desempenho do sistema. Neste sentido, o nível de atividade do SGBD poderia ser monitorado e as tarefas de sintonia agendadas para um momento de baixa atividade, a fim de que as ações de criação e reconstrução das estruturas de acesso tenham um impacto mínimo no desempenho do sistema. Para isso, pode-se armazenar um histórico do nível de atividades do SGBD e, com base neste histórico, prever qual o próximo momento no futuro em que o nível de atividades do SGBD estará abaixo do valor limítrofe. Esta funcionalidade pode ser implementada utilizando-se uma Rede Neural Artificial, por exemplo.

6. Adaptar e aplicar a abordagem proposta para sintonia automática de bancos de dados embarcados. Se as tecnologias de bancos de dados estão aptas a contribuir efetivamente com o gerenciamento e consulta de dados ubíquos, a sintonia automática também precisa ser ubíqua. Contudo, até este momento, poucos trabalhos existentes na literatura têm procurado possibilitar que bancos de dados “leves” (voltados para sistemas embarcados ou para aplicações distribuídas, tais como o Apache Derby, por exemplo) sejam auto-ajustáveis. Para isso é necessário identificar as principais características que fazem da sintonia automática uma abordagem interessante para estas plataformas e fornecer ferramentas que facilitem a aplicação das técnicas de sintonia automática para estes ambientes.
7. Conceber estruturas de índices mais adequadas para bancos de dados auto-ajustáveis. As estruturas de índices atualmente utilizadas em sistemas de bancos de dados (*B-Trees*, *R-Trees*, *Grid Files*), apesar de virem sendo utilizadas com sucesso há anos, não foram concebidas para serem ajustadas automaticamente. Neste sentido, é razoável considerar a possibilidade de mover a tarefa de sintonia de índices para um nível mais baixo, isto é, do nível de manutenção de uma configuração de índices para o nível físico (ou seja, de construção do índice). Se as estruturas de índices puderem, de alguma maneira, ajustar seu próprio uso, isto é, crescer ou decrescer de acordo com o número de acessos, isto poderia levar para um controle ou ajuste automático de granularidade muito mais fina. Pode-se pensar em estruturas de índices que sejam balanceadas a partir das características e/ou frequência do acesso aos dados e não com base na distribuição dos dados. Alguns trabalhos nesta linha já começaram a ser realizados (Sattler05, Paton07).