

1 Introdução

Esta dissertação discute mecanismos que possibilitam automatizar a recriação de índices fragmentados e analisa em detalhes uma heurística para tal automação. Durante a análise de comandos SQL, o otimizador eventualmente escolhe um determinado conjunto de índices que reduz custos de execução. Esta redução será influenciada por dois fatores: seletividade dos índices disponíveis e seus graus de fragmentação. Esta dissertação detém-se no segundo fator, discutindo mecanismos que reduzam seus malefícios. A seguir, comenta-se brevemente o contexto no qual está inserida esta pesquisa e apresentam-se resumidamente os próximos capítulos desta dissertação.

1.1.Contexto

Basicamente, o DBA (Database Administrator) desempenha duas atribuições cruciais: garantir a disponibilidade dos dados e assegurar níveis de desempenho aceitáveis. A primeira engloba tarefas tais como:

- Criação física de bancos de dados;
- Geração de objetos (tabelas, índices etc.);
- Proteção contra acessos indevidos;
- Manter políticas de cópias de segurança e recuperação, que garantam tempos mínimos de indisponibilidade dos dados.

Já a segunda tarefa relaciona-se com o grau de satisfação dos usuários em relação a tempos de respostas para comandos submetidos ao SGBDR, e, normalmente, resulta da correta aplicação de medidas prévias de ajuste, tarefa conhecida por sintonia [33]. Dada a permanente atividade em um banco de dados, realizar esta sintonia demanda a maior parte dos esforços de um DBA.

Diversos fatores afetam o desempenho de um SGBDR: há elementos externos, tais como equipamentos físicos (memória, processador, discos etc.), Sistema Operacional ou questões de conectividade. Também existem influências

internas que variam desde configurações em parâmetros, níveis de concorrência, aplicações e, principalmente, uma correta configuração de índices. [15] apresenta diversos casos onde o trabalho visando diminuir o esforço computacional de uma instrução SQL pode levar a ganhos consideráveis. Estes níveis dificilmente seriam alcançados realizando outras tarefas de Sintonia, tais como, alteração de parâmetros globais do SGBD, realização de ajustes no Sistema Operacional hospedeiro etc..

O Apêndice B desta dissertação apresenta uma pesquisa revelando a presença de ferramentas de apoio ao trabalho do DBA no que refere-se à busca por tempos de respostas menores. Entretanto, elas exigem trabalhos de configuração e limitam-se em apenas propor facilidades, ainda cabendo ao DBA a escolha da melhor alternativa. Desta forma, a maior parcela do trabalho de DBAs ainda depende bastante da experiência e conhecimento humanos. Dada à escassez de profissionais gabaritados, manter níveis de desempenho razoáveis torna-se um problema relevante [23].

Outro fator interessante a ser constatado advém da constante evolução dos Sistemas Gerenciadores de Bancos Relacionais, tais como Oracle, SQL Server, DB2, ou até mesmo os de menor porte, PostgreSQL e MySQL. Por exemplo, na versão 7.3 do SGBDR Oracle, era necessário configurar manualmente as alocações de espaços em disco dos segmentos (tabelas, índices etc.), bem como acompanhar seus crescimentos, e tomar medidas corretivas ao constatarem-se perigosos níveis de fragmentação. Este trabalho deixou de ser necessário nas versões subseqüentes, já que automatizaram-se as tarefas de alocação e manutenção de espaço. Desta forma, as necessidades de intervenções manuais por parte dos DBAs diminuem sensivelmente. Trata-se de uma tendência inexorável que proporcionará aos DBAs possibilidades de realizar trabalhos mais interessantes, como contribuir efetivamente na utilização produtiva dos dados ([21]).

Nesse contexto de dependência de mão de obra especializada em oposição às constantes melhorias em ferramentas, têm causado grande interesse, tanto no meio acadêmico, quanto no comercial, a necessidade de explorar a automação do processo de ajustes em um banco de dados (auto-sintonia). Diversos trabalhos corroboram esta idéia ([1], [17], [43]), que visa, a longo prazo, dispensar a ação manual de qualquer trabalho relativo à correção de desempenho.

Nossa dissertação concentra-se numa linha de estudos denominada Auto-sintonia de Índices”, que preconiza a realização de ajustes automáticos na execução de consultas SQL, visando reduzir-lhes tempos de resposta, a partir de alterações no conjunto de índices: criação, eliminação e recriação. Este trabalho teve como ponto de partida a dissertação de Marcos Salles [32], que seguiu a mesma linha, propondo um mecanismo automático de criação de índices. Nossa dissertação estende [32]: primeiro submetendo sua implementação a uma carga de trabalho alternativa gerada a partir do *benchmark* TPC-H, depois acrescentando o acompanhamento de índices criados, além de suas eliminações e reconstruções automáticas, levando em consideração níveis de preenchimento de páginas alternativos.

1.2.O Problema

Este trabalho de pesquisa aborda estruturas especiais denominadas índices, cuja finalidade consiste em acelerar buscas de dados. Suas presenças tornam-se obrigatórias, principalmente em tabelas de grande volume e muito consultadas. Já foi provado que o problema de seleção de índices é NP-difícil [7]. Portanto, não constitui surpresa que, apesar de grande esforço por parte dos DBAs, ainda existam sérios problemas de desempenho causados por configurações incorretas de índices.

Na discussão elaborada ao longo deste trabalho restringiram-se os tipos de índices utilizados àqueles que obedecem à estrutura em árvore conhecida como B+ (*btree+*). Trabalhos futuros poderiam estender idéias aqui apresentadas para que fossem utilizados outros tipos de índices.

Ainda que a utilização de índices seja um tópico extremamente importante, há muito desconhecimento sobre a necessidade de criá-los e, principalmente, de quando recriá-los. Por exemplo, suponha a existência da consulta mostrada na Figura 1.1, sobre uma tabela que sofra também muitas inserções e atualizações.

```

select sum(valor) from venda
where num between 8 and 64008
or num between 12800000 and 12864000
or num between 28800000 and 28864000
or num between 44800000 and 44864000
or num between 60800000 and 60864000;

```

Figura 1.1: consulta realizando cinco varreduras sobre uma tabela

Como os índices tratados nesta dissertação obedecem a organizações em árvore, um grande número de inserções, atualizações ou eliminações de chaves, fatalmente provocará divisão de páginas (*page splits*). Este fato pode causar fragmentação na estrutura, dependendo da localização onde venham a ser alocadas novas páginas. A Figura 1.2 apresenta um índice que não sofreu nenhum *page split*, enquanto que a Figura 1.3 exibe o mesmo índice após muitas atualizações.

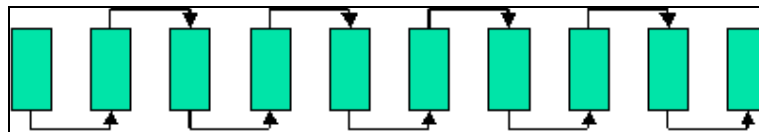


Figura 1.2: índice não fragmentado.

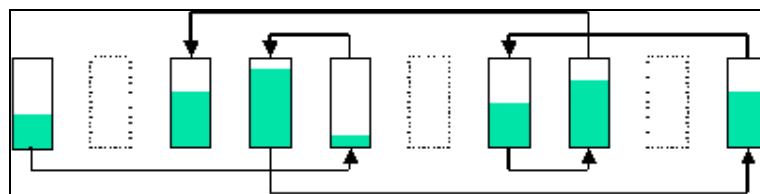


Figura 1.3: índice fragmentado

As setas apresentadas nas Figuras 1.2 e 1.3 revelam a ordem na qual o índice será percorrido. Deve-se perceber a ausência de “idas e voltas” em um índice apresentando grau de fragmentação zero. Já um índice fragmentado prejudica as operações de varredura, já que o artefato físico responsável pela leitura pode ser obrigado a realizar muitos deslocamentos físicos.

Uma solução imediata para esse problema seria a completa reconstrução periódica de todos os índices, de preferência em instantes de pouca atividade. Entretanto, tal abordagem traria dois graves inconvenientes:

1. Os grandes bancos de dados possuem dezenas de milhares de índices; Reconstruí-los todos de uma vez, por mais “quieto” que esteja o Sistema, causa um esforço computacional considerável;
2. Há índices, cujas tabelas não sofrem atualizações. Reconstruí-los não traz benefício algum.

Este trabalho consistirá em criar um ambiente onde índices sejam criados, destruídos ou recriados ao sabor dos comandos submetidos e levando em conta os malefícios causados pela fragmentação. Trata-se de uma abordagem adotada em [32], onde as operações sobre índices acontecem durante a realização da carga de trabalho. Assim como em [32], realizaram-se diversas implementações em um SGBD de código aberto, o PostgreSQL [31].

Em [28] elaborou-se um caso onde ficam evidentes os malefícios causados pela fragmentação. Mostram-se dados relativos a ocupação de espaço e duração de consultas, que atestam a necessidade em reconstruir índices periodicamente.

1.3 Estrutura da Dissertação

O restante da dissertação está organizado conforme descrito a seguir.

- O próximo capítulo relaciona vários trabalhos na área de Auto Sintonia, e resume as idéias discutidas em trabalhos de pesquisa que forneceram elementos para o desenvolvimento deste trabalho [32] e [26]. Também apresenta uma bateria de testes constituída por uma carga de trabalho diferente da utilizada em [32] e mostra quão eficaz foi a implementação de [32] realizando comparações com ferramentas que propõem índices em PostgreSQL, Oracle 10g e SQL Server 2005.
- O Capítulo 3 estende [32] propondo um mecanismo alternativo de eliminação automática de índices e conclui apresentando um caso onde a recriação de índices mostra-se interessante.
- O Capítulo 4 apresenta a Heurística de Reconstrução de Índices, responsável pela decisão quanto à reconstrução de um índice à beira da eliminação. Também apresentam-se detalhes de implementação, bem como diagramas seguindo a notação UML.

- O Capítulo 5 indica interessantes desdobramentos dos trabalhos aqui propostos e enumera as contribuições realizadas por esta dissertação.
- Há três apêndices: o primeiro (A) detalha o *benchmark* TPC-H, cujo esquema foi utilizado para gerar a carga alternativa utilizada no segundo capítulo; o Apêndice B mostra um relatório comparativo entre ferramentas de apoio aos trabalhos desempenhados por um DBA. Finalmente, o Apêndice C oferece um roteiro de acompanhamento das tarefas executadas pela ferramenta desenvolvida a partir das idéias presentes nesta dissertação.