

# 1 Introdução

## 1.1 Motivação

As aplicações de bancos de dados têm se tornado cada vez mais complexas e variadas. Atualmente, estas aplicações podem ser caracterizadas pelo grande volume de dados acessados e pela elevada demanda por desempenho, no que diz respeito tanto à diminuição do tempo de resposta das consultas quanto ao aumento da vazão (número de consultas executadas por unidade de tempo). Neste contexto, a sintonia do projeto físico de bancos de dados tem se revelado extremamente importante, uma vez que esta pode exercer um papel vital na melhoria do desempenho dos sistemas de bancos de dados.

O projeto físico inclui a seleção das estruturas de índices e visões materializadas a serem utilizadas, as tabelas a serem particionadas e os tipos de particionamento mais adequados, a duplicação de estruturas físicas, a replicação de dados e até mesmo a desnormalização de tabelas como forma de acelerar o desempenho da carga de trabalho. Infelizmente, realizar o ajuste do projeto físico de forma manual tem se tornado uma tarefa bastante complexa 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). A seguir, discutiremos brevemente as principais ações que compõem o projeto físico de bancos de dados.

A sintonia de índices, como parte do projeto físico de banco de dados, consiste na tarefa de selecionar, criar, excluir e reconstruir estruturas de índices com o objetivo de reduzir o tempo de processamento das cargas de trabalho. Dentre todas as atividades relacionadas à sintonia de bancos de dados, o ajuste das estruturas de índices é certamente uma das mais relevantes. Este fato decorre do grande benefício que estas estruturas podem trazer para o desempenho dos sistemas de bancos de dados, podendo reduzir substancialmente o tempo de execução das consultas, e até mesmo das atualizações (Sha03).

Essa redução decorre da diminuição da quantidade de páginas que necessitam ser acessadas (copiadas do disco para a memória principal) a fim de se obter a resposta para a consulta em questão. Entretanto, as estruturas de índices consomem espaço em disco, e, adicionalmente, podem ocasionar sobrecarga (“*overhead*”) na presença de atualizações, em razão da necessidade de se reorganizar estas estruturas.

Desta forma, para assegurar um bom desempenho à carga de trabalho submetida ao SGBD, torna-se de vital importância escolher um conjunto apropriado de índices (Monteiro06a). A tarefa de selecionar um conjunto de índices que seja adequado para uma determinada carga de trabalho, não é nada simples. Os SGBD’s atuais podem conter centenas de tabelas. Cada tabela pode ter diversos atributos. Um índice pode ser construído sobre uma ou mais colunas de uma relação. Logo, a quantidade de índices possíveis e/ou relevantes para uma determinada carga de trabalho pode ser muito grande.

Já foi provado que o problema de seleção de índices é NP-difícil (Cha04). Logo, não constitui surpresa o fato de que, apesar de grande esforço por parte dos DBAs (DBAs - *Database Administrators*), ainda existam sérios problemas de desempenho causados por configurações incorretas de índices.

Portanto, a sintonia de índices, quando possível, consome bastante tempo e exige administradores de bancos de dados experientes e especializados. Além disso, um conjunto de índices pode ser apropriado para uma determinada carga de trabalho e ser impróprio para uma outra carga de trabalho diferente. Outrossim, as cargas de trabalho submetidas ao banco de dados podem variar de forma sazonal, por exemplo, no começo de cada mês, estas cargas apresentam características OLPT (*On-Line Transaction Processing*), já no final do mês, em razão da emissão de relatórios gerenciais e sumarizados, as cargas de trabalho apresentam características OLAP (*On-Line Analytical Processing*). Isso dificulta ainda mais o já difícil desafio da seleção apropriada de índices (Monteiro06a).

Adicionalmente, a sintonia de índices pode ser bastante complexa em cenários dinâmicos, com análise exploratória e muitas consultas *ad-hoc*, onde a necessidade de índices não pode ser prevista. Um exemplo deste tipo de cenário são as aplicações OLAP (*Online Analytical Processing*) onde ferramentas voltadas para a inteligência de negócios (*business intelligence - BI*) e/ou ROLAP (*Relational Online Analytical Processing*) produzem consultas a partir de requisições de informações iniciadas pelo usuário (Bruno07a).

Outra importante tarefa relacionada ao projeto físico de bancos de dados consiste na duplicação de estruturas físicas. Atualmente, os SGBDs permitem que uma determinada estrutura física, uma tabela, por exemplo,

seja fisicamente ordenada (classificada) por um único critério. Este critério determina a ordem com que os registros da tabela são fisicamente armazenados em disco. Em geral, o critério de ordenação utilizado coincide com a chave do índice primário<sup>1</sup> definido sobre a tabela.

Desta forma, dada uma determinada tabela denominada **Venda**, não é possível ordená-la por dois critérios diferentes, data da venda e valor da venda, por exemplo. Contudo, a ordenação física dos registros que compõem uma tabela influencia fortemente o desempenho de determinadas consultas. Por exemplo, uma consulta agregada pela data da venda (**group by data**) apresentaria um tempo de execução menor se a tabela **Venda** estivesse fisicamente ordenada pelo atributo **data**, em comparação com a mesma tabela ordenada pelo atributo **valor**. Agora, suponha que existam duas consultas, igualmente importantes, uma agrupada pelo atributo **data** e outra agrupada pelo atributo **valor**. Neste caso, o DBA necessita escolher que consulta deve priorizar, uma vez que a tabela **Venda** somente pode ser ordenada fisicamente por um único critério. Uma possível solução para este problema, ainda pouco explorada na literatura, consiste na duplicação de estruturas físicas. Neste sentido, poder-se-ia duplicar fisicamente a tabela **Venda**, uma “cópia” seria fisicamente ordenada pelo atributo **data** e outra pelo atributo **valor**, com a finalidade de melhorar simultaneamente o desempenho das duas consultas discutidas. A duplicação de estruturas físicas será denominada de “*clusterização* alternativa de dados” e as “cópias” destas estruturas serão chamadas “*clusters* alternativos de dados”.

A manutenção de um conjunto adequado de visões materializadas também representa um aspecto de grande relevância no projeto físico de bancos de dados relacionais. A utilização de visões materializadas pode proporcionar expressivos ganhos de performance no processamento de consultas, especialmente em consultas agregadas sobre tabelas com grande volume de dados. Os resultados recentes do *benchmark* TCP-R mostram que o tempo de processamento da consulta pode ser melhorado por ordens de valor com o uso adequado de visões materializadas (Agra00).

As visões materializadas são relações derivadas, que são armazenadas como relações na base de dados. Usar visões materializadas para otimizar o processamento de consultas é uma idéia antiga, mas somente nos últimos anos esta idéia foi adotada em sistemas comerciais de bancos de dados.

Todavia, em virtude das restrições de espaço e custo de manutenção, a materialização de todas as visões possíveis torna-se inviável. Assim, um subconjunto das visões deve ser selecionado, a fim de maximizar a performance da carga de trabalho (*workload*) submetida ao SGBD. Entretanto, este também

<sup>1</sup>O conceito de índice primário é definido na Seção 2.2.

é um problema NP-Difícil (Agra00). Logo, a busca exaustiva torna-se uma solução inviável.

O particionamento de dados 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 (Papa06). Assim, o particionamento de tabelas e índices alivia o gerenciamento dos bancos de dados de grande porte, facilitando o gerenciamento em partes menores e mais manejáveis. No modelo relacional, o particionamento de dados consiste na divisão de uma relação em segmentos (fragmentos) menores através de um processo de seleção de *tuplas* (particionamento horizontal), projeção de colunas (particionamento vertical) ou uma combinação de ambas (particionamento híbrido ou misto).

Como pode ser observado, a manutenção do projeto físico é uma das tarefas mais complexas, e que consomem mais tempo, de um DBA. Contudo, esta não é a única atividade desempenhada por este profissional. As atividades do DBA envolvem o planejamento da capacidade de *hardware*, a garantia da segurança lógica dos dados, o projeto físico do banco de dados, o ajuste dos recursos gerenciados e o gerenciamento das dependências inter-sistemas (como, por exemplo, entre o *middleware* e o SGBD) (Monteiro06a). Conseqüentemente, fornecer ferramentas que auxiliem o DBA na difícil e repetitiva tarefa de ajustar o projeto físico de bancos de dados torna-se de fundamental importância.

Atualmente, os principais fabricantes de SGBDs (*Oracle, IBM, Microsoft*, dentre outros) oferecem ferramentas para suportar a sintonia do projeto físico de bancos de dados (Dias05, Zilio04, Bruno05). Estas ferramentas auxiliam os administradores (DBAs) através da análise automática da carga de trabalho e, com base nessa análise, recomendam a criação/remoção de índices, visões materializadas, índices definidos sobre visões materializadas e particionamento de grandes tabelas.

Contudo, tais ferramentas adotam uma abordagem *offline*<sup>2</sup> (ou seja, não executam de forma contínua) na solução do problema do projeto físico e transferem para o DBA a decisão final sobre os ajustes a serem realizados. Especificamente, os DBAs necessitam capturar uma amostra representativa da carga de trabalho e fornecer esta amostra para a ferramenta de sintonia. Além disso, os DBAs precisam adivinhar (supor) o momento em que o sistema necessita de ajustes, a fim de iniciar uma sessão de sintonia (executando explicitamente a ferramenta). Em seguida, os DBAs ainda necessitam decidir se devem ou não executar as recomendações sugeridas pela ferramenta, e,

<sup>2</sup>Capítulo 2.

principalmente, qual o melhor momento para executar estas recomendações, a fim de não comprometer o desempenho do sistema, uma vez que a criação de índices e/ou visões materializadas podem consumir preciosos recursos de processamento, por exemplo.

Adicionalmente, utilizar as ferramentas de sintonia fornecidas pelos fabricantes de SGBDs 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 melhoria de desempenho (Sal04).

Outra desvantagem das ferramentas de sintonia, também conhecidas como *advisors*, consiste na estratégia de analisar a carga de trabalho coletada durante um intervalo de tempo fixo e, com base nesta análise, criar, por exemplo, uma configuração de índices estática, ou seja, que não será alterada até que uma nova sessão de sintonia seja executada pelo DBA. Todavia, na maioria das aplicações recentes, a utilização do banco de dados muda com o tempo, isto é, as características da carga de trabalho submetida ao SGBD podem mudar (OLTP no início do mês e OLAP ao final do mês, por exemplo), o acesso aos dados pode variar de forma sazonal (apresentando picos de acesso em determinados períodos do dia ou do mês), novas aplicações que fazem acesso aos mesmos dados podem ser colocadas em produção, etc. Assim, a configuração adequadamente escolhida em determinado momento pode não mais ser adequada em um instante futuro (Sal04). Conseqüentemente, em ambientes dinâmicos, com várias consultas *ad-hoc*, torna-se bastante complicado identificar os índices potencialmente úteis, mesmo com o auxílio das ferramentas de sintonia. Assim, a sintonia do projeto físico não é um processo estático, mas, ao invés disso, consiste em um processo que necessita ser continuamente executado.

Pesquisas realizadas pelos principais fabricantes de SGBDs indicam que o gerenciamento autônomo do projeto físico de bancos de dados está ganhando grande aceitação como uma importante direção para o desenvolvimento de futuras ferramentas comerciais (Bruno07a, Cha07a). Recentemente, algumas iniciativas apresentaram descrições de protótipos que implementam funcionalidades na direção da sintonia automática (Sal04, Sal05, Cos05, Morelli06a, Sch06, Kai04, Luh07). Estes trabalhos, porém, adotam uma abordagem intrusiva (ou seja, exigem alterações no código fonte do SGBD) e funcionam apenas com um SGBD específico.

Nesta tese, no entanto, consideramos como uma característica especialmente importante o grau de independência da solução em relação ao código do SGBD (ou intrusividade). Neste contexto, podemos classificar as soluções

para o problema de sintonia automática como: Intrusivas e Não-Intrusivas. As soluções intrusivas são aquelas que exigem alterações no código do SGBD, ou seja, que estão fortemente acopladas ao código do SGBD. Conseqüentemente, funcionam apenas com o SGBD específico para o qual foram implementadas. Já as soluções não-intrusivas são aquelas que não requerem modificações no código do SGBD, ou seja, estão desacopladas deste código. Logo, as soluções não-intrusivas funcionam com qualquer SGBD e não são afetadas por alterações ou atualizações na implementação dos SGBDs. Entretanto, não encontramos na literatura nenhuma solução não-intrusiva e independente de SGBD para o problema da manutenção automática e *on-the-fly* (contínua) do projeto físico.

Por todas estas razões, acreditamos ser relevante investigar soluções para o problema da sintonia do projeto físico que sejam dinâmicas, autônomas, não-intrusivas e independentes de SGBD. Neste trabalho, propomos uma abordagem não-intrusiva, baseada em agentes de *software*, para o problema da manutenção automática e *on-the-fly* do projeto físico de bancos de dados. A abordagem proposta é completamente desacoplada do código do SGBD, pode ser utilizada com qualquer SGBD e executa independentemente de interações com usuários ou administradores (DBAs). Esta abordagem foi instanciada para solucionar dois importantes problemas relacionados ao projeto físico de bancos de dados: a manutenção automática de índices e a duplicação automática de estruturas físicas. Desta forma, comprovamos que a abordagem proposta pode ser instanciada e utilizada para a manutenção de diferentes estruturas que compõem o projeto físico. Além disso, propomos uma arquitetura baseada em agentes de *software* para suportar a manutenção automática do projeto físico de forma não-intrusiva. Especificamente, desenvolvemos um conjunto de heurísticas que executam continuamente e, sempre que necessário, modificam o projeto físico corrente, reagindo a alterações na carga de trabalho. O modelo de custos utilizado é independente de SGBD e considera o benefício das estruturas que compõem o projeto físico para a carga de trabalho submetidas ao SGBD. As técnicas desenvolvidas apresentam ainda, como características inéditas: o acompanhamento do nível de fragmentação das estruturas de índices, a manutenção conjunta de índices primários e secundários, além da definição e utilização dos conceitos de generalidade e relevância das estruturas de acesso, a fim de obter projetos físicos mais estáveis.

## 1.2

### Objetivo e Escopo

Neste trabalho, estamos interessados em um tipo particular de comportamento autônomo: a sintonia automática e não-intrusiva do projeto físico de bancos de dados. A principal questão investigada nesta tese é: será possível conceber um mecanismo que realize a sintonia do projeto físico de forma independente de SGBD? Indo ainda além, poderia esta solução:

- Ser não-intrusiva? Ou seja, ser completamente desacoplada do código do SGBD?
- Ser automática e *on-the-fly*?
- Este mecanismo poderia ser executado de forma contínua (ou periódica), durante o funcionamento normal (cotidiano) do SGBD?
- Ser completamente independente de interações com seres humanos?
- Ser independente de localização? Ou seja, a aplicação que executa os ajustes poderia executar em uma máquina distinta daquela utilizada para hospedar o SGBD?
- Ser suficientemente precisa, apenas de utilizar um modelo de custos genérico (independente de SGBD)? Teria esta solução resultados semelhantes ao de uma solução intrusiva e específica?

Esta tese mostra que é possível conceber uma estratégia para a manutenção automática do projeto físico que seja independente do SGBD utilizado e que não implique em alterações no código fonte dos SGBDs. Neste contexto, definimos uma abordagem não-intrusiva, baseada em agentes de *software*, para monitorar continuamente o funcionamento de um sistema de banco de dados e, sempre que necessário, modificar o projeto físico corrente de forma automática e *on-the-fly*, reagindo a alterações na carga de trabalho e procurando assegurar um desempenho sempre adequado, sem intervenção humana.

A abordagem proposta pode ser utilizada na manutenção automática das diferentes estruturas que compõem o projeto físico de bancos de dados relacionais, como, por exemplo:

- Índices;
- Visões materializadas;
- Particionamentos de tabelas;
- Estruturas físicas duplicadas;

Para comprovar a viabilidade das idéias discutidas, instanciamos a abordagem proposta para solucionar dois problemas fundamentais em projeto físico: a manutenção automática de uma configuração de índices e a duplicação automática de estruturas físicas (as quais denominamos *clusters* de dados), a fim de permitir duas ou mais ordenações físicas. Contudo, vale ressaltar que nossa abordagem pode ser aplicada na manutenção automática de qualquer componente do projeto físico. A abordagem apresentada é completamente desacoplada do código do SGBD, pode ser utilizada com qualquer SGBD e executa independentemente de interações com usuários ou administradores (DBAs). Desta forma, o trabalho exposto nesta tese pode ser classificado na linha de auto-sintonia local, contínua e não-intrusiva, sendo, portanto, a primeira solução não intrusiva para o problema da manutenção automática do projeto físico de bancos de dados.

As principais inovações trazidas pela abordagem proposta são: a utilização de um conjunto de heurísticas externas ao SGBD, as quais baseiam-se no conceito de otimização hipotética; a definição e utilização de um modelo de custos externo ao SGBD, ou seja, independente de SGBD; além da definição e utilização dos conceitos de relevância e generalidade das estruturas que compõem o projeto físico, com a finalidade de assegurar a qualidade e a estabilidade das configurações de projeto físico selecionadas. A instanciação da abordagem proposta para a manutenção automática de índices traz como inovação o acompanhamento do nível de fragmentação das estruturas de índices e a manutenção de índices primários e secundários. Já a instanciação da abordagem apresentada para a manutenção automática de estruturas físicas duplicadas é inovadora, uma vez que a duplicação de estruturas físicas (para permitir ordenações físicas distintas) como parte do projeto físico de bancos de dados ainda é um problema pouco explorado na literatura.

A fim de fornecer suporte para a abordagem proposta e permitir sua instanciação para diferentes problemas relacionados ao projeto físico, definimos uma arquitetura baseada em agentes de *software*. Os agentes que compõem a arquitetura foram implementados em linguagem Java. Estes agentes são responsáveis por desempenhar diferentes atividades: o monitoramento da carga de trabalho submetida ao SGBD, a seleção de estruturas de acesso, o gerenciamento de uma metabase local, a obtenção de informações estatísticas, a manutenção automática das estruturas de acesso que compõem o projeto físico, o agendamento das ações de sintonia (se necessário) e o gerenciamento das informações acerca da relevância das consultas, dentre outras.

### 1.3 As Principais Contribuições

As principais contribuições desta tese são:

1. Uma abordagem não-intrusiva para a manutenção automática do projeto físico de bancos de dados:

Esta abordagem é completamente desacoplada do código do SGBD em utilizado (não-intrusiva), podendo ser utilizada com qualquer SGBD, e executa independentemente de interações com os DBAs. Nossa abordagem segue o ciclo clássico de auto-sintonia: Observação, Predição e Reação, como descrito inicialmente em (Weikum94).

A abordagem proposta traz ainda como característica inovadora a utilização dos conceitos de “relevância”, “generalidade” e “estabilidade”, com o objetivo de assegurar a qualidade das configurações de projeto físico selecionadas.

2. Uma instanciação da abordagem proposta para o problema da manutenção automática das estruturas de índices. Esta instanciação apresenta como principais inovações:
  - Uma heurística integrada para a seleção e acompanhamento de índices (Heurística *HISAI*), a qual baseia-se no conceito de otimização hipotética. Essa heurística realiza, em um único passo, a seleção de índices candidatos (hipotéticos); o acompanhamento dos índices candidatos e reais; além do acompanhamento do nível de fragmentação dos índices reais.
  - A manutenção de índices primários e secundários: O mecanismo concebido para a manutenção automática de índices realiza a seleção e manutenção tanto de índices secundários quanto primários.
  - O acompanhamento do nível de fragmentação das estruturas de índices: A estratégia proposta realiza o monitoramento automático do nível de fragmentação das estruturas de índices fisicamente existentes. Logo que detectado que um índice se encontra num nível de fragmentação superior ao limite configurado (pelo DBA), este índice é automaticamente reorganizado. Assim, evita-se a ocorrência de problemas de desempenho advindos da existência de índices fragmentados.
  - Avaliação da 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.

3. Uma arquitetura baseada em Agentes de *Software* para a manutenção automática do projeto físico de bancos de dados: Esta arquitetura propõe a realização da sintonia automática do projeto físico por meio da colaboração entre agentes de *software*, os quais utilizam *drivers* para obter e manipular informações do SGBD utilizado. De posse desses dados, os agentes utilizam heurísticas que encapsulam o conhecimento de especialistas em sintonia de projeto físico, com a finalidade de manter um projeto físico sempre adequado.
4. Um modelo de custos externo: Este modelo de custos externo, ou seja, independente de SGBD, é utilizado pelas heurísticas concebidas e possibilita estimar, por exemplo, o custo de uma busca seqüencial, o custo de uma seleção utilizando um índice primário, etc.
5. Uma implementação da arquitetura e da abordagem proposta: A arquitetura e a abordagem não-intrusiva proposta foram implementadas inteiramente em linguagem Java. Este protótipo fornece suporte para os SGBDs *PostgreSQL*, *Oracle 10g* e *SQL Server 2005*. Contudo, o protótipo implementado pode ser facilmente estendido para outros SGBDs, uma vez que todas as informações necessárias para se instanciar os *drivers* que compõem a arquitetura de agentes são extraídas da metabase do SGBD utilizado (Monteiro08a).

## 1.4

### Organização da Tese

Esta tese está organizada da seguinte forma:

- No Capítulo 2 serão apresentados os conceitos básicos sobre indexação de dados, visões materializadas, particionamento de tabelas, projeto físico de bancos de dados, sintonia e auto-sintonia de banco de dados (definições, arquiteturas e características), *benchmarks*. Estas definições são necessárias para o desenvolvimento e entendimento do restante deste trabalho.
- No Capítulo 3 serão discutidas as principais abordagens, encontradas na literatura, para o projeto físico automático de bancos de dados. Uma análise comparativa detalhada destas abordagens também é apresentada. Além disso, justificamos e propomos uma nova classificação para as pesquisas em sintonia automática de bancos de dados.
- No Capítulo 4 apresentamos uma abordagem não-intrusiva para a manutenção automática do projeto físico de bancos de dados. Esta abor-

dagem é completamente desacoplada do código do SGBD utilizado, podendo ser utilizada com qualquer SGBD, e executa sem intervenção humana.

- O Capítulo 5 descreve como instanciar a abordagem proposta a fim de solucionar dois problemas clássicos do projeto físico de bancos de dados: a manutenção automática das estruturas de índices e a duplicação automática de estruturas físicas. Neste capítulo, a heurística integrada para a seleção e acompanhamento de índices (Heurística *HISAI*) é discutida e analisada. Um conjunto de algoritmos (gulosos e baseados em programação dinâmica) para seleção final de uma configuração de índices também é apresentado neste capítulo. A pré-criação de índices envolvidos em chaves primárias e estrangeiras é proposta e avaliada.
- No Capítulo 6 descrevemos uma arquitetura não-intrusiva para a manutenção automática do projeto físico de bancos de dados. Essa arquitetura propõe a realização da sintonia automática do projeto físico por meio da colaboração entre os agentes de *software*, os quais utilizam *drivers* para obter e manipular informações do SGBD utilizado. Cada um dos agentes e *drivers* que compõem a arquitetura são descritos. Além disso, discutem-se o funcionamento e as interações entre os componentes da arquitetura proposta.
- O Capítulo 7 discute os testes de desempenho realizados e os resultados obtidos.
- Finalmente, o Capítulo 10 conclui este trabalho, avaliando os resultados obtidos, as dificuldades encontradas e apontando direções para trabalhos futuros.

## 1.5

### Resumo do Capítulo

Este capítulo apresentou uma introdução ao problema da manutenção automática e *on-the-fly* do projeto físico de bancos de dados. As principais abordagens para solucionar este problema foram brevemente discutidas. O conceito de intrusividade, ou grau de independência de uma determinada abordagem em relação ao código do SGBD, foi definido. Com base nesta definição as soluções encontradas na literatura foram classificadas. Este capítulo discutiu, ainda, os objetivos, as principais contribuições e a estrutura da presente tese.

No próximo capítulo serão apresentados os principais conceitos necessários ao entendimento e desenvolvimento deste trabalho.