

Uma Arquitetura baseada em Agentes para a Resolução do Problema de Auto-sintonia de Índices em SGBDs Relacionais

Marcos Antonio Vaz Salles[✉], Sérgio Lifschitz (orientador)

Programa de Mestrado em Informática
Pontifícia Universidade Católica – Rio de Janeiro (PUC-Rio)
[mvsalles, sergio]@inf.puc-rio.br

Ano de Ingresso: 2002

Previsão de Conclusão: Março/2004

Resumo

A seleção de índices em bancos de dados relacionais é uma tarefa regularmente executada por DBAs para fazer com que uma determinada carga de trabalho seja processada pelo sistema com desempenho adequado. Neste trabalho, estudamos o problema de realizar a sintonia dos índices necessários para o sistema de forma totalmente automatizada. Isto envolve tanto reconhecer quais são as consultas que devem ser analisadas como também selecionar os índices adequados para estas consultas. Utilizamos uma arquitetura baseada em agentes para monitorar o SGBD e efetuar a criação e destruição de índices quando necessário. As características de autonomia e de adaptabilidade de agentes de software facilitam a tarefa de tomar decisões de modificação sobre os índices existentes diante de mudanças na carga de trabalho processada pelo SGBD. Em relação a outros trabalhos presentes na literatura, pretendemos nos diferenciar pela derivação automática das consultas e frequências componentes da carga de trabalho para aplicações OLTP que podem possuir características de sazonalidade e pela possibilidade de criação e destruição de índices totalmente sem intervenção de um administrador.

[✉] Trabalho financiado pela FAPERJ – Fundação de Amparo a Pesquisa do Rio de Janeiro.

1. Introdução

A sintonia de bancos de dados (*database tuning*) é a arte de ajustar os parâmetros, configurações e estruturas de um sistema de banco de dados às necessidades de uma determinada carga de trabalho. Por carga de trabalho entendemos um conjunto de consultas ou atualizações ponderadas por suas frequências de ocorrência.

Os atuais produtos de bancos de dados oferecem entre dezenas e centenas de parâmetros e configurações que podem ser ajustados para alcançar um funcionamento mais eficiente. O ajuste destes parâmetros é complexo, tanto pela necessidade de compreensão dos algoritmos que são usados pelo sistema quanto pela possibilidade de interrelações entre os ajustes. Pode-se dizer que na tarefa de ajuste de bancos de dados um bom princípio é "Pensar globalmente, consertar localmente" [20].

Desta forma, é necessária uma quantidade crescente de profissionais especializados para realizar o ajuste de desempenho das complexas aplicações de bancos de dados hoje existentes. Como podemos perceber, à medida que os bancos de dados vão se tornando cada vez mais comuns em nossa sociedade, esta dependência de recursos humanos se torna muito indesejável.

Isto levou tanto o meio acadêmico quanto a indústria a colocar como um item de grande prioridade em suas pautas de pesquisa a criação de sistemas de bancos de dados que sejam capazes de auto-sintonia e de auto-administração [12, 2]. Espera-se que, no futuro, os sistemas no mínimo necessitem de uma quantidade menor de profissionais especializados para a sua administração.

As iniciativas de pesquisa se encaixam, até o momento, em duas grandes frentes: criar sistemas que sejam auto-sintonizáveis por projeto (veja [5]) ou atacar problemas de sintonia dos sistemas atualmente disponíveis para torná-los mais autônomos. Na segunda frente de pesquisa, vários problemas podem ser considerados, como o ajuste do nível de concorrência do sistema, o posicionamento de dados ou o uso mais adaptativo de memória [21].

O nosso trabalho de pesquisa se foca no **problema de auto-sintonia de índices em bancos de dados relacionais**. O problema é uma versão estendida do *problema de seleção de índices em bancos de dados relacionais*. Este último consiste em, dada uma carga de trabalho $W = \{(Q_i, f_i), i=1, \dots, n\}$ em que Q_i é uma consulta SQL e f_i é a sua frequência de submissão, minimizar o custo total de processamento de W pelo sistema respeitando um limite de espaço S disponível para a criação de índices. O custo total de processamento é a soma dos custos de acesso, atualização e manutenção dos índices. Já foi demonstrado que uma versão restrita do problema de seleção de índices é NP-difícil [6].

Um das questões práticas envolvidas durante a seleção de índices é a obtenção da carga de trabalho de entrada. Isto raramente é uma tarefa simples. Assim, definimos o problema de auto-sintonia de índices em bancos de dados como o problema de automaticamente determinar a carga de trabalho e os índices adequados para esta carga de trabalho de forma a minimizar o consumo de recursos do sistema. Resolver o problema de auto-sintonia de índices em bancos de dados efetivamente significa eliminar da agenda do DBA toda e qualquer preocupação com a criação ou destruição de índices. No nosso trabalho, iremos focar nossa atenção no processamento de aplicações OLTP, possivelmente com características de sazonalidade. Um exemplo de carga de trabalho sazonal seria a execução mensal de um processo de contabilização em um sistema bancário de contas correntes.

O restante deste documento está organizado conforme descrito a seguir. A seção 2 discute diversas propostas apresentadas na literatura para lidar com os problemas de seleção de índices e de auto-sintonia de índices. A seção 3 discute o estado atual da pesquisa

sendo desenvolvida para lidar com estes problemas. Por fim, a seção 4 mostra alguns dos resultados esperados e contribuições trazidas pelo trabalho de pesquisa.

2. Trabalhos Correlatos

Existem várias propostas na literatura que atacam o problema de seleção de índices em bancos de dados. As primeiras tentativas de solução procuraram utilizar técnicas de modelagem analítica para caracterizar as estatísticas a serem coletadas pelo sistema e os critérios de decisão para criação dos índices. Conforme pode ser visto em [11], a modelagem formal do problema mesmo com o escopo reduzido a consultas e atualizações envolvendo uma única relação encontra diversas dificuldades, tanto na qualidade das estatísticas a serem acumuladas quanto na própria complexidade do modelo criado.

Uma segunda geração de técnicas propõe que sejam criadas ferramentas que se utilizem do próprio otimizador de custos do SGBD para realizar a avaliação da qualidade dos índices testados. Temos como exemplos de técnicas desta geração as propostas [8] e [9]. Um conceito que passa a ser utilizado é o de índice virtual ou hipotético. Um índice virtual é um índice considerado pelo otimizador para a geração de um plano de acesso, mas que existe apenas no catálogo do SGBD e não fisicamente. As propostas se concentram, então, em criar módulos com estratégias de busca para a enumeração de configurações de índices hipotéticos que são custeadas pelo otimizador do SGBD. A melhor configuração hipotética encontrada é recomendada para o usuário da ferramenta ou criada após um período de observação configurado no SGBD.

Uma generalização natural do problema de seleção de índices é o problema de projeto físico de bancos de dados. Em [19], é apresentado um *framework* para a seleção de estruturas de arquivos aplicáveis a uma determinada carga de trabalho. O trabalho propõe que modelemos cada estrutura de arquivo possível como uma *característica* que pode ser aplicada ao esquema do banco de dados. A estratégia de busca deve enumerar as *características interessantes* para as consultas presentes na carga de trabalho e avaliar se o uso destas características produz planos de menor custo para a execução conjunta destas consultas.

Como o problema de seleção de índices é relevante tanto do ponto de vista acadêmico quanto do ponto de vista da indústria, alguns dos principais fornecedores de bancos de dados investiram recursos no aprimoramento de ferramentas para a recomendação de índices. Em [1, 3, 4], é descrita a ferramenta de seleção de índices implementada para o Microsoft SQL Server. É dada uma carga de trabalho como entrada para a ferramenta com n consultas SQL. Esta carga de trabalho é particionada em n cargas de trabalho que são analisadas independentemente. É proposto um algoritmo guloso para realizar a análise dos índices candidatos para cada consulta SQL que consegue enumerar índices de múltiplas colunas até um determinado limite. Depois de selecionar os melhores índices para cada carga de trabalho, os índices propostos são considerados como candidatos para a carga de trabalho inicial como um todo. Reforçamos que a única modificação feita no otimizador de consultas nesta proposta é a capacidade de levar em consideração índices virtuais no processo de otimização. Esta ferramenta também consegue sugerir a criação de visões materializadas.

As propostas anteriores utilizam o otimizador para avaliar possíveis conjuntos de índices, mas nunca para recomendar os índices que devem ser empregados. Em [14], temos a proposta de uma ferramenta para seleção de índices para o IBM DB2. Podemos encaixar esta proposta já em uma possível terceira geração de ferramentas para seleção de índices, uma vez que ela prega uma integração muito mais íntima com o otimizador de consultas do banco de dados. É sugerido que o próprio otimizador do SGBD tenha a capacidade de enumerar os melhores índices para uma dada consulta SQL. Para isto, todos os índices virtuais relevantes são enumerados antes do processo de otimização e fica totalmente a cargo do otimizador

selecionar quais índices serão efetivamente utilizados. Com esta capacidade embutida no otimizador, somente resta o problema de achar os melhores índices para uma dada carga de trabalho (conjunto de consultas SQL). Este problema é encarado como uma variante do Problema da Mochila e é proposta uma heurística gulosa para a sua resolução.

O Oracle também possui uma ferramenta para recomendação de índices, que faz parte do Oracle Enterprise Manager 9i Tuning Pack [17]. Até onde vai o nosso conhecimento, não foi publicada a estratégia de busca empregada por esta ferramenta para calcular quais índices devem ser criados ou removidos após a determinação da carga de trabalho.

Todas as propostas discutidas acima têm como uma de suas principais dificuldades a definição de uma carga de trabalho relevante (ou, de forma equivalente, a definição de um período de observação do sistema). Desta forma, a qualidade do projeto físico obtido depende fundamentalmente da carga de trabalho submetida pelo DBA para a ferramenta. Isto se distancia do ideal de um sistema que selecione os seus próprios índices automaticamente.

A proposta de [7] procura atacar esta deficiência. Os autores mostram como o uso da tecnologia de agentes de *software* pode ser útil para a construção de SGBDs que se auto-sintonizam e se auto-administram. É analisada em detalhes a aplicação deste tipo de tecnologia ao problema de auto-sintonia de índices. É proposto um agente de software integrado ao código do SGBD e construído em camadas. Este agente coleta informações sobre as consultas submetidas ao SGBD e, com base em uma heurística, decide quando deve criar ou destruir índices em benefício do sistema. Infelizmente, esta proposta de agente não foi implementada para que se pudesse avaliar a qualidade dos índices gerados e a penalidade sobre o desempenho do sistema.

3. Estado Atual da Pesquisa

No presente momento, ainda não realizamos a defesa da proposta de dissertação para a pesquisa. Estamos conduzindo um estudo detalhado do estado da arte para as áreas de auto-sintonia de bancos de dados e de seleção de índices. Já vislumbramos, entretanto, algumas direções em que podemos proceder com o trabalho, o que nos leva a acreditar que em breve realizaremos a defesa de proposta.

Um primeiro ponto a ser explorado vai de encontro a uma das principais deficiências das demais propostas existentes: a falta de uma base de código publicamente disponível para se analisar a integração entre o módulo de seleção de índices e o otimizador do SGBD. Para atacar este problema, estamos realizando a implementação da proposta de [7] em um banco de dados de código fonte aberto.

Uma alternativa seria o uso de um sistema de banco de dados relacional implementado com conceitos de agentes de software sendo desenvolvido na própria PUC-Rio. Como este sistema ainda não se encontra disponível, entretanto, optamos em proceder com nosso trabalho de implementação no SGBD MySQL [16]. Mesmo sendo um produto relativamente limitado em comparação com outros bancos de dados comerciais, o banco de dados MySQL já apresenta todos os rudimentos necessários para uma otimização de comandos baseada em custos. Precisaremos trabalhar sobre o código do sistema, porém, para estender as rotinas de otimização de forma a poder levar em consideração índices virtuais.

A proposta de [7] se baseia no uso de agentes de software para atacar o problema de auto-sintonia de índices em bancos de dados. Advogamos que o uso de agentes se adequa muito bem aos problemas de sintonia de bancos de dados, uma vez que agentes podem ser construídos de forma a possuir características de autonomia, adaptabilidade e inteligência. Um estudo de como agentes de software podem ser utilizados em conjunto com SGBDs foi realizado em [15]. O nosso trabalho de pesquisa é, na verdade, uma extensão dos trabalhos

realizados em [7] e [15]. Estendemos o trabalho de [7] por procurarmos de fato implementar uma arquitetura de agentes para a escolha de índices. Já o trabalho de [15] é estendido por estarmos investigando uma aplicação de agentes para aumentar as capacidades de um SGBD.

Estamos utilizando na implementação uma arquitetura de agente em camadas, proposta originalmente em [13]. Até o momento, nossos esforços de implementação têm sido muito facilitados por esta escolha, uma vez que as camadas podem ser testadas de forma incremental. Outra característica interessante da arquitetura utilizada é o seu uso intensivo de padrões de projeto [10], o que torna a sua implementação em uma linguagem orientada a objetos como C++ muito mais limpa e simples.

Mesmo com o uso de agentes, atacar o problema de coleta de uma carga de trabalho relevante para a análise e criação de índices sem definir uma classe de aplicações de estudo é inviável. Nossos estudos se concentram em aplicações OLTP, possivelmente com comportamentos sazonais, e na criação ou destruição de índices convencionais (árvores B+, implementadas pela maioria dos SGBDs comerciais). Esta direção de pesquisa ainda precisa ser melhor desenvolvida.

Por fim, observamos que uma outra área em que a pesquisa pode trazer contribuições é no projeto e implementação de heurísticas para o problema de seleção de índices. Inspirados no tratamento dado em [14], podemos investigar o projeto de novas heurísticas para o problema através do uso de metaheurísticas [18]. Metaheurísticas, como GRASP, melhoria iterativa e algoritmo genético, são arcabouços para o projeto de heurísticas eficientes para a resolução de problemas NP-difíceis.

Para este trabalho de implementação de heurísticas, acreditamos que utilizar uma implementação baseada em agentes também trará benefícios. Podemos concentrar nossas alterações somente na estratégia de busca codificada na camada de raciocínio do agente, mantendo as demais camadas responsáveis pela integração com o SGBD intocadas.

4. Resultados Esperados

Esperamos que a pesquisa tenha como principais contribuições:

- apresentar claramente a separação entre os problemas de seleção de índices e de auto-sintonia de índices em bancos de dados;
- detalhar uma arquitetura de agentes de *software* voltada para o problema de auto-sintonia de índices em bancos de dados;
- obter resultados experimentais da implementação de técnicas para a resolução dos problemas de seleção de índices e de auto-sintonia de índices em um SGBD de código fonte aberto, como é o caso do MySQL.
- investigar a possibilidade de proposta de novas heurísticas para o problema de seleção de índices através do uso de metaheurísticas;

Por fim, é importante observar que o trabalho de pesquisa pode ter um desdobramento futuro interessante através da produção de uma ferramenta comercial de auto-sintonia de índices. Esta possibilidade existe uma vez que, conforme visto na seção 2, os principais SGBDs comerciais já possuem capacidades de fazer a avaliação de custos de consultas na presença de índices hipotéticos.

Referências

- [1] Agrawal, S.; Chaudhuri, S. and Narasayya, V., Automated Selection of Materialized Views and Indexes for SQL Databases, Procs 26rd VLDB Intl Conference, 2000, pp 496--505.
- [2] Bernstein, P.; Brodie, M.; Ceri, S.; Dewitt, D.; Franklin, M.; Garcia-Molina, H.; Gray, J.;

Held, J.; Hellerstein, J.; Jagadish, H.; Lesk, M.; Maier, D.; Naughton, J.; Pirahesh, H.; Stonebraker, M. and Ullman, J., The Asilomar Report on Database Research, ACM SIGMOD Record 27(4), 1998, pp 74--80.

[3] Chaudhuri, S. and Narasayya, V., AutoAdmin "What-if" Index Analysis Utility, Procs ACM SIGMOD Intl Conference on Management of Data, 1998, pp 367--377.

[4] Chaudhuri, S. and Narasayya, V., Microsoft Index Tuning Wizard for SQL Server 7.0, Procs ACM SIGMOD Intl Conference on Management of Data, 1998, pp 553--554.

[5] Chaudhuri, S. and Weikum, G., Rethinking Database System Architecture: Towards a Self-tuning RISC-style Database System, Procs 26rd VLDB Intl Conference, 2000, pp 1--10.

[6] Comer, D., The Difficulty of Optimum Index Selection, ACM Transactions on Database Systems 3, 1978, pp. 440--445.

[7] Costa, R.L.C. and Lifschitz, S., Index Self-Tuning and Agent-Based Databases, XXVIII Latin-American Conference on Informatics (CLEI), 12pp. CD-ROM Procs, pp 76 Abstracts Procs., 2002.

[8] Finkelstein, Schkolnick, M. and Tiberio, P., Physical Database Design for Relational Databases, ACM Transactions on Database Systems 13, 1, 1988, pp. 91--128.

[9] Frank, M.; Omiecinski, E. and Navathe, S., Adaptive and Automated Index Selection in RDBMS, Procs Intl Conference on Extending Data Base Technology, 1992, pp 277--292.

[10] Gamma, E.; Helm, R.; Johnson, R. and Vlissides, J., Design Patterns: Elements of Reusable Object-Oriented Software, Addison Wesley, 1994.

[11] Hammer, M., Chan, A., Index Selection in a Self-Adaptive Data Base Management System, Proceedings of the ACM SIGMOD International conference on Management of data, Washington, D.C., EUA, 1976, pp 1--8.

[12] P. Horn, 2001. Autonomic Computing: IBM's Perspective on the State of Information Technology, disponível em <http://www.research.ibm.com/autonomic/manifesto>.

[13] Kendall, E., Krishna, P., Murali, P., Pathak, C. and Suresh, C.B., A Framework for Agent Systems, chapter in Implementing Application Frameworks: Object-Oriented Frameworks, ed. Fayad, M., Schmidt, D.C. and Johnson, R., Wiley & Sons, 1999.

[14] Lohman, G., Valentin, G., Zilio, D., Zuliani, M., Skelley, A., DB2 Advisor: An Optimizer Smart Enough to Recommend Its Own Indexes, Proceedings of the 16th IEEE Conference on Data Engineering, San Diego, CA, 2000, pp. 101--110.

[15] Macêdo, J., Agent-based DBMSs study, MSc thesis, PUC-Rio – Departamento de Informática. September 2000 (in portuguese).

[16] <http://www.mysql.com> .

[17] Oracle Tuning Pack: feature overview, disponível em http://otn.oracle.com/products/oem/pdf/TP_9iR2_FO.pdf

[18] Rayward-Smith, V. J., Modern heuristic search methods, Wiley & Sons, 1996.

[19] Rozen, S. and Shasha, D.; A Framework for automating Physical Database Design, Procs 17th VLDB Conference, 1991, pp 401--411.

[20] Shasha, D. and Bonnet, P., Database Tuning: Principles, Experiments and Troubleshooting Techniques, Morgan Kaufmann, 2003.

[21] Weikum, G.; Hasse, C.; Mönkeberg, A. and Zabback, P., The Confort Automatic Tuning Project, Information Systems 19(5), 1994, pp 381--423.