Abordagens para a pesquisa por palavras-chave em base de dados estruturadas

André Santos[1] e Sérgio Nunes[2]

[1] Portugal Telecom Inovação,
Aveiro, Portugal
andre-m-santos@ptinovacao.pt
[2] INESC Porto / DEI,
Faculdade de Engenharia, Universidade do Porto
Porto, Portugal
sergio.nunes@fe.up.pt

Resumo Os sistemas de pesquisa na *web* tornaram popular a pesquisa por palavras-chave. Este método de pesquisa melhorou claramente a usabilidade das pesquisas na *web*. Por outro lado, as tradicionais pesquisas estruturadas que conhecemos da área das bases de dados estruturadas envolvem um formulário com várias caixas de texto que testam determinados atributos na base de dados. Deste modo, tem havido um esforço crescente nos últimos anos, tanto a nível académico como a nível empresarial, para a criação de aplicações que permitam a pesquisa por palavras-chave sobre bases de dados estruturadas. Por um lado, surgiram extensões nos sistemas de gestão de base de dados relacionais para dar resposta a esta necessidade, e por outro, apareceram vários sistemas open-source baseados em modelos não relacionais. Este trabalho compara as capacidades reais das plataformas existentes que seguem ambos os modelos para resolver eficientemente a pesquisa por palavras-chave em informação estruturada.

1 Introdução

Os tradicionais formulários de pesquisa associados às base de dados relacionais apresentam um conjunto de caixas de texto para preenchermos os critérios de pesquisa. No entanto, este método de pesquisa não é tão conveniente como a pesquisa por palavras-chave que é implementada nos motores de busca mais conhecidos da Internet (Google, Yahoo, Bing) sobre informação não estruturada. Com apenas uma caixa de texto, o utilizador ao inserir os termos da pesquisa obtém instantaneamente sugestões de resultados. Assim, neste trabalho foi feita uma comparação de duas abordagens para a pesquisa livre, não estruturada, sobre bases de dados relacionais.

Inspirado pelo grande sucesso obtido nos sistemas de pesquisa na *web*, a pesquisa por palavras-chave em base de dados estruturadas emergiu como um tópico de investigação que tem tido bastantes contribuições [1] nos últimos anos. Em contraste com este modelo de pesquisa, nas pesquisas estruturadas são indicadas explicitamente as restrições a aplicar a cada atributo, o que permite

que os critérios de pesquisa possam ser directamente traduzidos numa consulta SQL. Na pesquisa por palavras-chave, devido à subjectividade dos termos da pesquisa, não há o conceito de resposta válida à consulta. Assim, pretende-se obter resultados que sejam o mais próximo possível dos critérios da pesquisa [1].

Os Sistemas de Gestão de Base de Dados mais populares fornecem mecanismos para o suporte de pesquisa por texto. As implementação dos índices de texto destes sistemas consiste numa variante do índice invertido. Na sua representação básica, um índice invertido é constituído por um vocabulário, conjunto distinto de palavras-chave, onde cada palavra guarda um apontador para o início de uma lista invertida que armazena um conjunto de referências para os tuplos da base de dados que a contêm a palavra-chave e o número total desses documentos [2].

Existem vários projectos open-source focados no problema da pesquisa por palavra-chave. Numa comparação realizada por Middleton e Baeza-Yates [3] destacam-se dois, o Apache Solr [5] e o Sphinx Search [4]. O Apache Solr tem como vantagens a maturidade do projecto, a rapidez da pesquisa e a possibilidade de configuração. A plataforma Sphinx Search destaca-se na eficiência, na relevância dos resultados e na simplicidade de integração com bases de dados relacionais.

2 Problema

O objectivo deste trabalho é a definição de um sistema que, partindo de informação estruturada, permita a pesquisa de grandes volumes de informação de forma não estruturada. Tendo por base a melhoria de usabilidade nos formulários de pesquisa, os requisitos deste projecto também passam pela independência do esquema de base de dados e a eficiência na consulta de grandes volumes de dados.

Um sistema de base de dados permite a execução de inúmeras operações diferentes, que vão desde as operações para criação do esquema de base de dados (criação de tabelas, índices e chaves) ao CRUD sobre a informação armazenada nas relações. No âmbito dos sistemas da PT Inovação duas das operações do CRUD têm uma frequência muito elevada: “inserir” e “pesquisar” e as outras duas: “actualizar” e “apagar” ocorrem com uma frequência bastante menor. Se tomarmos em consideração um sistema que faz a gestão da informação de uma empresa de telecomunicações, as operações de tributação das chamadas, recarga dos telemóveis e consulta de saldo têm relevância máxima e um grau muito reduzido de tolerância. Estes três processos são constituídos por operações de inserção e consulta. Deve-se ainda realçar que estas duas últimas operações são executadas na maior parte das vezes em tempo real, o que eleva ainda mais o grau de exigência perante operações executadas em situações de manutenção, como as operações de “actualizar” e “apagar”.

3 Metolodogia

Para concretizar o objectivo deste trabalho, foi decidido criar uma bateria de testes baseada nos casos de uso discutidos no Secção 2 que colocasse à prova

um conjunto de tecnologias escolhidas de entre aquelas que foram abordadas na Secção 1.

O esquema de dados utilizado foi inevitavelmente condicionado por Apache Solr, que é um sistema orientado ao "documento", onde não existe o conceito de normalização originado nas base de dados relacionais. Decidiu-se então definir um esquema de base de dados desnormalizado, que pode ser visto como uma única relação com um vasto conjunto de atributos.

Cada teste às plataformas corresponde à execução de um determinado número de vezes da aplicação cliente, onde em cada execução a aplicação efectua uma tarefa dependente do caso de uso a que se refere o teste. A aplicação deve registar as latências de execução da tarefa, entre outros valores dependentes do teste. Os testes criados para os três casos de uso foram estruturados de forma diferente.

4 Resultados Experimentais

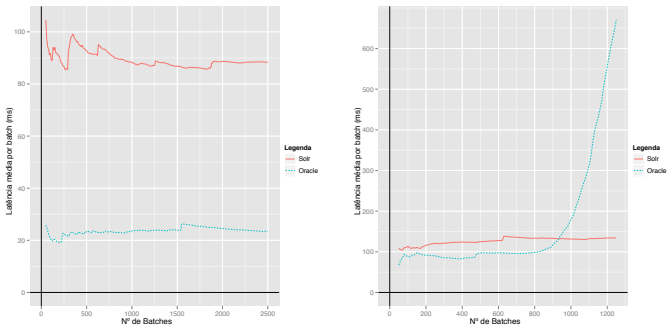
Este capítulo descreve os resultados experimentais obtidos com o trabalho efectuado seguindo a metodologia apresentada na Secção 3.

4.1 Caso de uso "inserir"

Neste caso de uso o objectivo é confirmar a tendência de Solr e Oracle a acumular informação na base de dados. A informação é inserida em *batches* para as bases de dados, cada um contendo um conjunto fixo de registos. A Figura 1 apresenta-nos a relação do aumento do número de *batches* inseridos na base de dados com o tempo médio de inserção de cada *batch*. Cada ponto P(**x**,**y**) do gráfico indica que o tempo médio de inserir e indexar **x** *batches* consecutivos é de **y** milisegundos. Através da observação do gráfico do lado esquerdo, onde é indexado 1 campo de texto em 20 possíveis, verifica-se que as curvas estabilizam nos 89 milisegundos no caso do Solr e nos 23 milisegundos no caso do Oracle, aproximadamente a partir dos 1500 *batches* inseridos.

Ao indexar 10 campos de texto é beneficiada a plataforma Solr como se comprova pelo gráfico do lado direito da Figura 1. Solr tem um desempenho muito bom na indexação de grandes conjuntos de atributos como podemos observar ao longo deste caso de uso. Naturalmente surge uma degradação considerável de performance em Oracle, mas quanto ao Solr, as latências aumentam 50% no global relativamente ao cenário anterior, o que é um aumento pouco significativo. Neste trabalho Oracle, tal como Solr, é utilizado com a sua configuração de base. Por outro lado, Oracle é considerado um sistema muito completo e com um grande poder de configuração. Aumentando as exigências a nível de indexação, Oracle exige que a sua configuração seja bastante refinada. Neste caso, considerando a curva exponencial de Oracle, conclui-se que necessita claramente de uma reconstrução ou optimização dos índices. A nível de espaço, ambas as plataformas criam estruturas de dados com dimensões muito semelhantes.

Ao utilizar índices de texto, em vez dos índices por omissão das plataformas temos resultados mais claros. Os resultados do teste podem ser consultados na



(a) Indexando 1 campo de texto (b) Indexando 10 campo de texto

Figura 1: Latência média de inserção dos *batches* à medida que a informação acumula na base de dados.

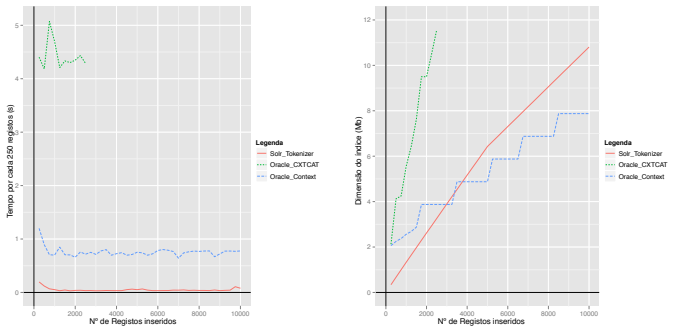
Figura 2. Neste caso temos uma superioridade muito clara do Solr desde o início do teste. De salientar que o teste feito ao índice CTXCAT de Oracle terminou bastante mais cedo que os outros índices, uma vez que a taxa de inserção de dados era extremamente lenta. A nível da dimensão dos índices ao longo da indexação, destaca-se novamente o mau desempenho do índice CTXCAT. Nos outros dois tipos de índice a diferença não é muito significativa, no entanto, o índice CONTEXT optimiza mais a estrutura que cria, o que contrasta com o facto de apresentar latências médias das consultas aproximadamente 12 vezes superiores às de Solr.

4.2 Caso de uso "pesquisar"

Neste teste pretende-se averiguar como variam as latências das consultas às bases de dados com a quantidade de informação que armazenam. A Figura 3 representa os gráficos gerados neste teste, no lado esquerdo, utilizando os índices por omissão das plataformas e no lado direito, utilizando índices de texto. Neste teste as plataformas foram preparadas para indexar 10 campos de texto.

No gráfico do lado esquerdo é notório um desfasamento entre as latências de pesquisa do Oracle e do Solr. Indexando um campo de texto, o desempenho entre os dois sistemas é muito semelhante, no entanto ao indexar 10 campos de texto o gráfico do Oracle escala claramente na vertical e, por seu lado, o gráfico do Solr mantém-se muito semelhante. É ainda notória uma degradação das latências do Oracle acompanhada pelo acumular de registos na base de dados, aproximadamente a partir dos 220 mil registos. Em Solr, a tendência ao acumular de registos é do aumento das registos, ainda que muito ligeiro.

No gráfico do lado direito, temos comportamentos mais claros. As latências de pesquisa em Oracle são claramente influenciadas ao utilizar um índice de texto,



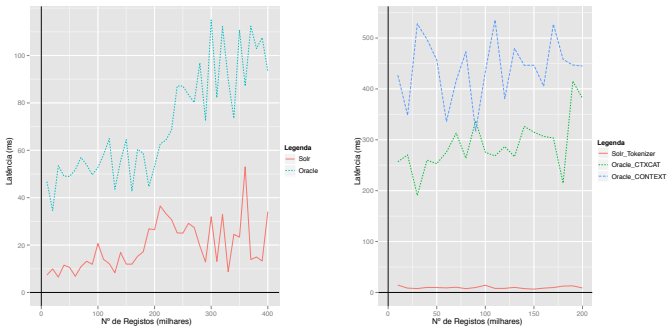
(a) Análise do tempo (b) Análise do espaço

Figura 2: Análise de latências das consultas e dimensão das estruturas de dados criadas à medida que a base de dados acumula informação, utilizando índices de texto.

seja o índice Oracle CTXCAT ou o índice Oracle CONTEXT. Em contraste com as pesquisas utilizando o índice por omissão B-Tree, temos um aumento das latências das pesquisas do intervalo 40 - 100 ms para o intervalo 200 - 500 ms, o que é muito significativo quando comparado com Solr. Este último regista tempos de resposta na pesquisa por palavras muito semelhantes aos registados na pesquisa por frases, apresentados no gráfico do lado esquerdo, onde utilizar um índice que não filtra o texto por espaços. Deste modo, Solr mostra que o seu comportamento é muito pouco influenciado quando é configurado para indexar e pesquisar palavras.

5 Conclusões

Para definir um sistema que permita a pesquisa por palavras-chave em base de dados estruturadas, foram estudadas plataformas que se baseiam em modelos diferentes: uma solução relacional e outra não relacional. O método utilizado para averiguar as capacidades das plataformas consistiu na execução de uma bateria de testes que cobria os casos de uso mais relevantes do perspectiva da empresa PT Inovação. O poderoso sistema relacional evidenciou uma clara necessidade de que a sua configuração seja bastante adaptada às especificidades do ambiente onde vai executar. Solr, um sistema baseado num modelo orientado ao "documento", mostrou-se muito vocacionado para a indexação ao apresentar um desempenho praticamente inalterado com a variação da quantidade e complexidade da informação a indexar. Deste modo, propõe-se a coexistência das duas plataformas para aproveitar a capacidade de armazenamento, interrogação e processamento de um sistema relacional como Oracle e a vocação para a in-



(a) Utilizando os índices por omissão (b) Utilizando os índices de texto

Figura 3: Latência média de cada consulta à medida que as plataformas acumulam informação, utilizando índices normais e de texto.

dexação de um sistema que seja baseado num modelo diferente, como o Apache Solr.

O trabalho realizado analisa as possibilidades de adoptar soluções baseadas ou não em modelos relacionais para dar resposta à pesquisa por palavras-chave em informação estruturada. Como trabalho futuro, poderia ser estendida a plataforma de testes criada em várias direcções, nomeadamente aprofundando os testes ao registar indicadores como a quantidade de memória principal e de CPU utilizado e alargando a gama de plataformas a analisar.

Reconhecimentos Agradecemos ao Mestrado Integrado em Engenharia Informática e Computação da Faculdade de Engenharia da Universidade do Porto pelo apoio financeiro para a publicação e apresentação deste trabalho.

Referências

1. L. Fang, C. Yu, W. Meng, A. Chowdhury: Effective Keyword Search in Relational Databases. Em: SIGMOD'06 (2006)
2. D. Harman, E. Fox, R.A. Baeza-Yates, W. Lee: Inverted files. Em: Information Retrieval: Data Structures and Algorithms (1992)
3. C. Middleton, R. Baeza-Yates: A Comparison of Open Source Search Engines. Relatório técnico, Universitat Pompeu Fabra (2007)
4. Sphinx Search - Open Source Search Server, `http://sphinxsearch.com/`
5. Y. Seeley: Apache Solr. Apachecon Europe (2006)
6. J. Zhang, S. Wang: ITREKS: Keyword Search over Relational Database by Indexing Tuple Relationship. Em: DASFAA'07 (2007)