

Avaliação do desempenho do uso de Sistemas Gerenciadores de Bancos de Dados Orientados a Objetos para armazenamento de dados multimídia

Bruno Eustáquio P. Marques ¹, Ivre Marjorie R. Machado ¹,
Poliana A. C. de Oliveira ¹

¹ Instituto de Informática – Pontifícia Universidade Católica de Minas Gerais
(PUC Minas) Caixa Postal 1.686 – 30.640-070 – Belo Horizonte – MG – Brasil

{brunoppl, poliana.correa}@gmail.com, ivre@pucminas.br

RESUMO

Esse artigo apresenta um estudo comparativo entre dois modelos de bancos de dados com intuito de avaliar se os bancos de dados orientados a objetos são uma alternativa viável para substituir os bancos de dados relacionais em aplicações que trabalham com dados multimídia. Para os testes foram escolhidos quatro SGBDs: MySQL, PostgreSQL, Neodatis e Db4o. A medição em um ambiente que simula condições de uso real foi o método adotado para a avaliação de desempenho. Os resultados mostraram que é possível usar um SGBDOO para este tipo de aplicação. Contudo, ainda falta um amadurecimento maior deste modelo para que ele possa atingir o nível do paradigma predominante no mercado, o SGBDR.

ABSTRACT

This article introduce a comparative study between two types of databases in order to assess whether the database object-oriented are a viable alternative to replace relational databases with applications that work with multimedia data. For the four tests were chosen SGBDs: MySQL, PostgreSQL, Neodatis and Db4o. Measurement in an environment that simulates conditions of actual use was the method adopted for the evaluation of performance. The results that it is possible to use a SGBDOO this type of application. However, there is still a greater maturity of this model so that it can reach the level of the prevailing paradigm in the market, the SGBDR.

1. INTRODUÇÃO

A computação é uma área em constante mudança e nos últimos tempos está cada vez mais presente no cotidiano das pessoas e organizações. De um modo geral, sistemas computacionais geram muitos dados. Para lidar mais facilmente com estes tipos de dados, que a princípio compreendiam textos e números, foi criado o modelo de Sistema Gerenciador de Banco de Dados Relacional (SGBDR).

Entretanto, na década de 90 houve uma mudança substancial na demanda de armazenamento de dados. Com o advento da Internet e das aplicações multimídia os SGBDR começaram a lidar com as dificuldades impostas pela complexidade de armazenamento e recuperação de dados multimídia tais como vídeos, imagens e sons.

Os Sistemas Gerenciadores de Bancos de Dados Orientados a Objetos (SGBDOO) surgiram durante a década de 80 com a expectativa de criar um modelo de banco de dados baseado no mesmo paradigma das linguagens de programação orientadas a objetos como Java, Smalltalk e C++. Era esperado que esse novo modelo fosse capaz de substituir o modelo relacional. Porém, isto não se concretizou devido a forte consolidação deste último no mercado.

Apesar de não serem tão populares, os SGBDOO apresentam algumas vantagens em relação aos SGBDR. Eles utilizam objetos para representar entidades do mundo real e apresentam uma melhor capacidade para tratamento de objetos com maior nível de complexidade como os dados de conteúdo multimídia. Essa característica leva a seguinte hipótese: seriam eles uma alternativa viável para a substituição de SGBDR no armazenamento e gerenciamento de dados multimídia? A partir dessa pergunta surge o problema que será tratado por este artigo.

Este artigo está organizado da seguinte forma. A seção dois apresenta alguns conceitos teóricos. A seção três cita alguns trabalhos relacionados a este. Já a seção quatro descreve como se dará a avaliação proposta por este trabalho. A seção cinco mostra os resultados obtidos após a execução dos testes. Por fim, a seção seis apresenta uma conclusão e sugere alguns trabalhos futuros.

2. REFERENCIAL TEÓRICO

Navathe e Elmasri (2005, p4) definem Sistema Gerenciador de Banco de Dados (SGBD) como uma coleção de programas que permitem a um usuário gerenciar e manter um banco de dados. Ainda segundo estes autores, um SGBD é “*um sistema de software de finalidade genérica que facilita o processo de definição, construção, e manipulação de bancos de dados para várias aplicações*”.

Dentre os modelos de SGBD que surgiram nas últimas décadas, de acordo com Silberschatz et al. (2006, p.25), o modelo relacional é o paradigma predominante nos sistemas comerciais de processamento de dados devido a sua simplicidade. Silberschatz et al. (2006, p241) afirmam ainda que, como os SGBDR ficaram cada vez mais populares e passaram a ser usados em uma variedade maior de aplicativos e, como estes continham tipos de dados mais elaborados, algumas limitações deste modelo

começaram a surgir.

Como uma alternativa ao modelo relacional que possibilitava lidar com tipos de dados mais complexos surgiram os Sistemas Gerenciadores de Bancos de Dados Objetos-Relacionais (SGBDOR). Silberschatz et al. (2006, p.241) mencionam alguns obstáculos enfrentados pelos programadores ao usar este modelo. O primeiro deles está relacionado com as limitações de tipos de dados necessários em aplicações complexas. Além disso, existem diferenças entre os tipos usados pelas linguagens orientadas a objetos e os implementados pelos SGBDs o que torna uma dificuldade adicional para eles.

Com a expectativa de criar um modelo de dados adequado ao paradigma da orientação a objetos, conforme Greene (2006), surgiram os Sistemas Gerenciadores de Bancos de Dados Orientados a Objetos (SGBDOO). Acreditava-se que esse novo modelo fosse capaz de substituir o modelo relacional, que era o mais utilizado até então. No entanto, essa expectativa não chegou a se concretizar nos anos seguintes, pois o modelo relacional já estava bem consolidado.

Com relação aos dados multimídia, Kalipsiz (2000), os conceituam como uma representação de diferentes tipos de informação (texto, imagens, objetos gráficos, vídeo e áudio) na forma de sinais digitais. De acordo com Silberschatz et al. (2006, p.619), o dado multimídia é um tipo cada vez mais popular e que normalmente fica armazenado em sistemas de arquivos. O armazenamento em arquivos não costuma ser um problema quando o número de objetos não é muito grande. Entretanto, se este número aumentar são necessários recursos de um SGBD, tais como: atualizações transacionais, facilidades para consulta, capacidade de indexação, entre outros.

3. TRABALHOS CORRELATOS

Foram analisados três trabalhos com assuntos relacionados ao tema estudado. No primeiro deles Silva (2009) compara o desempenho entre três formas de recuperação de dados multimídia em um banco de dados *PostgreSQL*. Ele concluiu que o método que apresenta melhor desempenho é o armazenamento através de referência externa, com a ressalva de que este método não recupera o arquivo em si, mas apenas a sua localização no disco.

Já o trabalho de Paterson et al. (2006) apresenta uma comparação entre vários SGBDR e o *Db4o* que é um SGBDOO. Foram usados o *Pole Position* e um *Benchmark* para medir o desempenho de tratamento de objetos nos bancos de dados. Durante os testes, os SGBDs avaliados foram submetidos a uma mesma carga de trabalho e realizadas medidas de tempo de cada operação. O autor concluiu que o acesso de dados navegando por objetos em um SGBDOO se mostra mais adequado. Entretanto, no caso de consultas por dados simples uma consulta com SQL apresenta melhor resultado.

Por fim, a dissertação de Andrade (1998) mostra um levantamento das principais tecnologias envolvidas na utilização de SGBDs Multimídia apresentando um estudo que leva em consideração a complexidade dos dados Multimídia e como isto reflete na arquitetura de um SGBD. Ao final do trabalho ele apresenta um protótipo de um Sistema de Informação que lida com dados Multimídia e conclui que SGBDOO são indicados para este tipo de aplicação.

4. PROPOSTA

A proposta deste artigo compreende a definição de métricas e execução de testes a fim de verificar o desempenho de SGBDOO e SGBDR para armazenar e gerenciar dados multimídia. O objetivo principal é determinar qual dos dois modelos é o mais adequado para esta aplicação. Para realizar este estudo comparativo foram utilizados dois SGBDOO (*Db4o* e *Neodatis*) e dois SGBDR (*MySQL* e *PostgreSQL*). A escolha do *MySQL* e *PostgreSQL* se deve à popularidade desses sistemas no mercado, além de possuírem uma grande comunidade, serem gratuitos e utilizados em vários tipos de aplicação. O *Db4o* e o *Neodatis* foram selecionados também por serem gratuitos e pela facilidade de encontrar materiais na *Web* sobre os mesmos.

4.1 Caracterização do ambiente de testes

A máquina utilizada para a execução dos testes possui as seguintes características: Processador Athlon X2 7750 2,9 Ghz; 1,5 GB de memória RAM 667 Mhz; HD SATA 160 GB 7200 RPM e Sistema operacional Ubuntu Linux 9.04-32bits.

Os testes foram realizados em uma instalação padrão do Ubuntu 9.04 acrescido de um ambiente de desenvolvimento Java (Java SDK 1.5 e IDE Eclipse 3.5) além dos SGBDs: *Db4o*, *Neodatis*, *MySQL* e *PostgreSQL*.

A coleção de dados multimídia utilizada como carga de trabalho é composta de imagens de diferentes resoluções e arquivos de áudio com diversas taxas de qualidade. Para os testes, os arquivos do tipo imagem foram separados em pastas contendo: 10, 50, 100, 500 e 1000 arquivos, com tamanhos variando entre 7KB e 3.8MB, nos formatos JPG e PNG. A Tabela 1 apresenta o número de registros e o tamanho das pastas utilizadas. É importante observar que a pasta com 500 arquivos apresenta um tamanho superior à pasta de 1000 arquivos. Isto acontece, pois ela contém imagens de resolução maior.

Tabela 1. Pastas de imagens usadas no teste

Numero de Registros	Tamanho Total
10	1,7 MB
50	14,6 MB
100	66,9 MB
500	683,5 MB
1000	203,4 MB

Para realizar os testes com os arquivos de áudio foram criadas pastas contendo 10, 20, 50, 100 e 200 arquivos de áudio em formato MP3, com tamanhos variando entre 1.3 MB e 30 MB. O número de registros e o tamanho das pastas utilizadas são mostrados na Tabela 2.

Tabela 2. Pastas de arquivos de áudio usadas nos testes

Numero de Registros	Tamanho Total
10	63,5 MB
20	263,2 MB
50	464,7 MB
100	536 MB
200	942,7 MB

Após a preparação do ambiente de testes foram definidas algumas métricas, tendo em vista, a comparação dos SGBDs.

Com relação ao desempenho as principais características observadas foram:

- O tempo gasto para realizar uma série de operações;
- A escalabilidade, ou seja, a capacidade de manipular um aumento na carga de trabalho de maneira uniforme;
- Consumo de espaço utilizado em disco.

Para a execução dos testes foi elaborado um aplicativo em Java com as seguintes funcionalidades:

- Efetuar conexão com o banco de dados;
- Listar um diretório de arquivos na máquina e inseri-los no banco de dados;
- Recuperar dados multimídia armazenados no banco de dados;
- Realizar a medição do tempo gasto para a realização destas operações.

Cada SGBD utilizado possui uma maneira diferente de efetuar o armazenamento de um arquivo binário. A fim de tentar manter a maior igualdade possível entre as condições de uso dos SGBDs avaliados adotou-se o mesmo método de armazenamento para todos: conversão dos arquivos a serem armazenados em um *array* de *bytes*.

Para testar a integridade dos dados após a conversão do arquivo em sequências de *bytes* foi adicionado um método ao aplicativo com a função de recuperar a sequência de *bytes* do SGBD, realizar a sua conversão em um arquivo de imagem e, em seguida, exibi-lo na tela.

É importante destacar que todos os testes foram executados por três vezes sob as mesmas condições e os resultados apresentados aqui representam a média dos valores obtidos em cada caso.

Conforme mostra a próxima seção, durante os testes o *Neodatis* apresentou um comportamento discrepante em relação aos outros SGBDs. Ele ocupou um espaço em disco bem desproporcional se comparado aos arquivos armazenados além de ter gasto mais tempo para executar as operações. Por este motivo, a avaliação do *Neodatis* tornou-se inviável frente aos outros SGBDs

1.RESULTADOS

5.1 Inserção de arquivos de imagem

O primeiro teste realizado foi a inserção de arquivos de imagem nos SGBDs avaliados. Os resultados podem ser vistos na Tabela 3.

Tabela 3. Inserção de imagens (Espaço em disco em MB)

Inserção de arquivos de imagem (Espaço em disco em MB)				
Qtde	MySql	PostgreSql	Db4o	Neodatis
10	1,71	24	1,7	32,6
50	14,6	38	14,6	227,5
100	66,95	91	67	
500	683,53	689	683,6	
1000	203,4	231	203,5	

Observe que passou a ser inviável a realização de testes no *Neodatis* após 100 arquivos. A Tabela 3 mostra que nos testes iniciais com até 50 arquivos, o *Neodatis* apresentou um comportamento inesperado. O espaço utilizado por ele para armazenamento de arquivos foi bem superior aos demais SGBDs.

Já os outros SGBDs testados apresentaram comportamentos semelhantes, na maioria dos casos. O *MySQL* e o *Db4o* criaram um arquivo com praticamente o mesmo tamanho que as imagens inseridas. E o *PostgreSQL* criou um arquivo um pouco maior que o das imagens inseridas, mas nada muito expressivo.

A Tabela 4 mostra o tempo gasto para inserção de arquivos de imagem nos SGBDs.

Tabela 4. Inserção de imagens (Tempo gasto em ms)

Inserção de arquivos de imagem (tempo gasto em ms)				
Qtde	MySql	PostgreSql	Db4o	Neodatis
10	586	1241	629	4453
50	1705	3240	2802	38300
100	6384	13806	9353	
500	74434	127780	69248	
1000	28008	67128	39191	

Assim como no teste anterior, o *Neodatis* apresentou um comportamento inesperado. O *MySQL* e o *Db4o* apresentaram desempenho semelhante neste quesito, com uma leve vantagem para o *MySQL*. O *PostgreSQL*, por sua vez, se mostrou significativamente mais lento que os outros dois SGBDs para essa tarefa.

5.2 Inserção dos arquivos de áudio

O segundo teste realizado foi a inserção das pastas com arquivos de áudio. Esse teste é parecido com a inserção de arquivos de imagem no que diz respeito a operação realizada nos SGBDs, a principal diferença está no tamanho dos arquivos inseridos.

Tabela 5. Inserção de arquivos de áudio (Tempo gasto em ms)

Inserção de arquivos de áudio (tempo gasto em ms)				
Qtde	MySql	PostgreSql	Db4o	Neodatis
10	4029	8968	5850	235953
20	19095	43151	19775	
50	34224	70408	37880	
100	41174	90233	48001	
200	97849	187573	97839	

Conforme mostra a Tabela 5, com um comportamento semelhante aos testes de inserção de imagens, mais uma vez o *MySQL* foi o mais ágil. O *Db4o* também apresentou um bom desempenho, já o *PostgreSQL* novamente foi o mais lento dentre os SGBDs testados.

O *Neodatis* não consta no gráfico devido ao fato de ser inviável testá-lo para esta tarefa. Ao tentar a inserção de 10 arquivos de áudio ele gastou quase 4 minutos contra os tempos inferiores a 10 segundos dos outros SGBDs. Além disso, quanto ao espaço em disco o *Neodatis* utilizou quase 1GB para armazenar apenas 63,5MB de arquivos, conforme ilustra a Tabela 6.

Tabela 6. Inserção de arquivos de áudio (Espaço em disco MB)

Inserção de arquivos de áudio (espaço em disco em MB)				
Qtde	MySQL	PostgreSQL	Db4o	Neodatis
10	63,5	88	63,5	1015
20	263,16	292	263,2	
50	464,67	496	464,7	
100	536,04	555	536,1	
200	942,76	971	942,8	

Observa-se também que no teste de espaço em disco os SGBDs apresentaram resultados semelhantes com uma pequena desvantagem para o *PostgreSQL*. O *MySQL* e *Db4o* tiveram praticamente um empate ao mostrar o menor espaço em disco.

5.3 Recuperação de uma imagem

Este teste consistiu em medir o tempo de recuperação de uma imagem no SGBD. Os SGBDRs fizeram a recuperação do arquivo através de uma consulta SQL pelo nome do arquivo em questão. O *Db4o* executou a consulta através da comparação de um atributo nome de um dado objeto com todos os objetos armazenados na base de dados. O objeto ao ser recuperado deveria retornar uma cadeia de *bytes* e, a partir dela, a imagem seria reconstituída. A Tabela 7 mostra os tempos gastos por eles para realizar a tarefa descrita.

Tabela 7. Recuperação de uma imagem (Tempo em ms)

Recuperação de um registro (tempo gasto em ms)			
MySQL	PostgreSQL	Db4o	Neodatis
613	367	833	

Neste teste não houve um resultado tão uniforme quanto nos testes anteriores. O *PostgreSQL* que levou desvantagem nas tarefas de inserção de arquivos, foi o mais rápido na recuperação de dados. O *MySQL* foi o segundo colocado e o *Db4o* apresentou o pior resultado consumindo mais do que o dobro do tempo gasto pelo *PostgreSQL*.

2.CONCLUSÃO E TRABALHOS FUTUROS

Os testes apresentados demonstraram que os SGBDR obtiveram os melhores resultados, sendo que o *MySQL* teve o melhor resultado para a inserção de registros e o *PostgreSQL* o melhor para execução de consultas. Apesar disso, foi possível constatar que um SGBDOO pode ser usado em uma aplicação que trabalhe com dados multimídia.

Mesmo com a impossibilidade de executar todos os testes usando o *Neodatis*, o *Db4o* apresentou bons resultados com relação ao desempenho para a inserção de registros e utilização de disco. O ponto fraco deste SGBDOO em relação aos SGBDR foi o tempo gasto para a consulta de um determinado registro.

O *Db4o* pode não ter tido os melhores resultados neste caso, mas foi capaz de apresentar conceitos interessantes. Um deles é a eliminação da necessidade de conversão de um modelo de classes utilizado na programação para um modelo implementado pelo SGBD, o que facilita para o programador que pode persistir os objetos de sua aplicação sem ter que usar intermediários para isso.

Como trabalhos futuros um possível foco a ser explorado é a análise do comportamento de SGBDR e SGBDOO em sistemas mais complexos e talvez uma nova comparação dos SGBDR e

SGBDOO utilizando as ferramentas específicas de cada SGBD para armazenar dados multimídia. Além de uma análise mais aprofundada sobre o processamento de consultas que envolvem dados multimídia.

7. REFERÊNCIAS BIBLIOGRÁFICAS

- [1] Andrade, Nelson Spangler. 1998. *Sistemas de informação multimídia*. Dissertação de mestrado, Universidade Federal de Minas Gerais. DOI= <http://www2.dcc.ufmg.br/laboratorios/npdi/orientacoes/mestrado/spangler/dissertacao.pdf> (Ago.2010).
- [2] Elmasri, R.; Navathe, S. 2005. *Sistemas de banco de dados*. 4ª.ed. São Paulo: Addison-Wesley.
- [3] Greene, R. 2006. *OODBMS Architectures*. DOI= <http://www.odbms.org/experts.aspx#article> (May. 2009).
- [4] Kalipsiz, Oya. 2000. *Multimedia Databases*. Fourth International Conference on Information Visualisation IV. London.
- [5] MySQL. 2009. *Reference Manual*. DOI= <http://dev.mysql.com/doc/refman/5.4/en/index.html>. November.
- [6] Neodatis. 2009. *Neodatis Object Database*. DOI= http://sourceforge.net/projects/neodatis_odb/files/NeoDatis%20ODB%20for%20Java/1.9/1.9.20/neodatis-odb.1.9.20.662.zip/download. November.
- [7] Paterson, J.; Edlich, S.; Hörning, H. 2006. *The definitive guide to Db4o*. New York: Apress.
- [8] PostgreSQL. 2009. *Comprehensive Manual*. DOI= <http://www.postgresql.org/files/documentation/pdf/8.3/postgresql-8.3-A4.pdf>.
- [9] Silberschatz, A.; Korth, H.; Sudarshan, S. 2006. *Sistema de Banco de Dados*. 6ª.ed. Rio de Janeiro: Campus.
- [10] Silva, R. C. 2009. *Benchmark em Banco de Dados Multimídia: Análise de Desempenho em Recuperação de Objetos Multimídia*. DOI= http://dspace.c3sl.ufpr.br:8080/dspace/bitstream/1884/4683/1/disserta_ricardo.pdf.