

Desempenho do Uso de Visões nas Consultas em Diferentes Servidores de Banco de Dados Relacionais

Walter José da Silva¹, Hiago de Assis Silva¹, Flávio Ferreira Borges²

¹Universidade Federal de Goiás – Campus Jataí
BR 364, Km 193, nº 3800, CEP: 75801-615 – Jataí – GO – Brasil

²Departamento de Ciências da Computação
Universidade Federal de Goiás – Campus Jataí – Jataí, GO – Brasil
{sillva.walter, hiagoassis, flavio.ufg.jatai}@gmail.com

Abstract. *This paper presents a research on the performance of queries using view in the database, to identify whether the use of this feature can bring benefits to alleviate the overload of requests for database servers, reducing the response time. And also which will show the performance gain from using this feature for different database management systems in order to know if there is a performance difference and see which of the servers tested support better the feature of view.*

Resumo. *Este artigo apresenta um trabalho de investigação do desempenho do uso de visão em consultas ao banco de dados, visando identificar se o uso desse recurso pode trazer benefícios para amenizar a sobrecarga de requisições em servidores de banco de dados, diminuindo o tempo de resposta. E também mostrar qual será o ganho de desempenho do uso desse recurso para diferentes sistemas gerenciadores de banco de dados, a fim de saber se há uma diferença de desempenho e verificar qual dos servidores testados suporta melhor o recurso de visão.*

1. Introdução

A evolução das organizações e o aumento no volume de dados manipulados ocasionaram o surgimento de uma série de sistemas específicos para tratar as várias necessidades de um banco de dados. Dentre essas necessidades destacam-se o controle do acesso aos dados, o aperfeiçoamento do desempenho e a garantia da integridade. Para facilitar esta administração surgiu o Sistema Gerenciador de Banco de Dados (SGBD) [Silberchatz *et al.* 2006].

O SGBD é considerado como uma coleção de programas que permite aos usuários a criação e manutenção de um banco de dados, portanto, é um sistema de *software* de propósito geral que facilita os processos de definição, construção, manipulação e compartilhamento de um conjunto de dados entre vários usuários e aplicações [Elmasri *et al.* 2005].

O meio de acesso às informações mantidas por esse SGBD é conhecida por *query*, que é simplesmente uma forma de requisitar informações ou puramente dados ao banco. Essa forma de requisitar informações é possível graças a um padrão criado no

início da década de 70, chamado de SQL (*Structured Query Language*) [Date 2003]. A SQL é uma linguagem para lidar com bancos de dados relacionais, e que abrange instruções para definição de dados, consultas e atualizações, dentre outras funcionalidades [Date 2003], [Elmasri *et al.* 2005], [Ramakrishnan *et al.* 2008].

Porém, essas requisições consomem certo tempo de processamento, e em sistemas de grande porte, onde o número de requisições simultâneas é elevado, o servidor de banco de dados pode encontrar dificuldades em enviar as repostas aos processos chamadores (clientes) em um tempo hábil. E para amenizar a sobrecarga dos servidores no processamento dessas requisições sugere-se o uso de visão em consultas ao banco de dados.

Em banco de dados, visão é um recurso avançado onde algumas de suas funções são fornecer um meio de abreviação ou de macro, permitir que vários conjuntos de dados (tabelas) sejam vistos por usuários em um novo conjunto, fornecer segurança automática para dados ocultos e independência de dados lógica [Date 2003]. E que segundo Date (2003), Elmasri (2005) e Halevy (2001), trazem um ganho significativo no tempo de resposta da consulta.

Contudo, o objetivo deste trabalho é verificar o desempenho em tempo de resposta das consultas com o uso do recurso de visão e compará-lo ao desempenho de consultas sem o uso desse recurso, para comprovar de fato a sua eficiência. E também comparar o desempenho de visões em diferentes SGBDs, a fim de mostrar se há uma diferença relevante no tempo de resposta entre eles.

Para avaliar o desempenho das consultas com o recurso de visão de SGBDs é usado à técnica de *benchmark*. E conforme Ciferri (1995) essa técnica é baseada em um modelo de análise, composta por um conjunto de testes já previamente definidos e que poderá ser aplicado a vários sistemas distintos, que possuam funcionalidades semelhantes. O objetivo é manter toda a avaliação de desempenho dentro das mesmas características. O JMeter é ferramenta usada neste trabalho, e possui essas técnicas mencionadas.

O Apache JMeter é uma aplicação *desktop*, projetada para testar e medir o comportamento do desempenho funcional de aplicações cliente/servidor, como aplicações *web*, banco de dados ou aplicativos de FTP (*File Transfer Protocol*). É uma das ferramentas (*open-source*) amplamente utilizada em testes de sistema, oferecendo à aplicação de testes em rede livremente distribuída. É puramente baseado em Java e é altamente extensível através de uma API (*Application Programming Interface*) fornecida pela Apache [Halili 2008], [Santos *et al.* 2008].

Todavia, autores como Pires *et al.* (2006), Silva (2006) e Aquino (2007) mostram a importância de avaliar recursos ou certas funcionalidades em sistemas computacionais, que nesse caso específico são servidores de banco de dados. O motivo é verificar se há diferenças de desempenho em funcionalidades de sistemas, visando encontrar a ferramenta certa para uma aplicação específica.

Enfim, tem-se visto que essa investigação será importante para obtenção de métodos que diminuirão um problema tão comum em dias atuais, onde a geração de informação armazenada cresce demasiadamente, e a busca por essas informações chega a ser frenética. Técnicas para amenizar a sobrecarga de servidores de dados são

importantes, e o recurso de visão pode ser mais uma dentre outras várias técnicas existentes a realizar essa tarefa.

2. Trabalhos Relacionados

O objetivo que se tem com esses trabalhos é apresentar as discussões e problemas relacionados ao uso de visão em consultas ao banco de dados, e encontrar formas e métodos de se fazer uma avaliação de desempenho entre *softwares* distintos. De forma que essa avaliação seja realizada em um alto nível de imparcialidade, buscando o embasamento necessário para mostrar a relevância de se fazer essa análise.

Começando com o trabalho de Mitra (2001) onde é apresentado um algoritmo que permite que um sistema que manipula uma quantidade muito grande de informações possa reescrever consultas de forma eficiente utilizando visão. Com isso, reduzir o tempo de resposta aos usuários, melhorar a escalabilidade e tornar o sistema capaz de lidar com um grande número de fontes de informação¹.

Nos trabalhos de Halevy (2001) e Felea *et al.* (2007) é mostrado a importância de usar visão em consultas ao banco de dados. É dito que visão é uma funcionalidade que ajuda a resolver uma série de problemas de gerenciamento de dados. A otimização da consulta, a manutenção da independência física dos dados, a integração de dados e o projeto de *design* de armazenamento de dados são os problemas citados por Halevy (2001), que podem ser resolvidos por meio do uso de visão.

Halevy (2001) ainda ressalta que as muitas aplicações da visão usadas para responder consultas estimularam uma onda de pesquisa, que foram desde os fundamentos teóricos até o projeto do algoritmo e a sua implementação em vários sistemas comerciais. E Felea *et al.* (2007) reuni pontos de vista de vários autores para falar sobre os problemas que foram solucionados com o uso de visão.

O problema em servidores de banco de dados sobrecarregados é um problema abordado por Kyoung-Don *et al.* (2007), ele fala principalmente em aplicações de tempo real onde o tempo de resposta para uma requisição é um fator crítico. Com base em estudos apresentados por autores já citados, o uso de visão em consultas ao chamado RTDB (*Real-Time Database*), poderia ser uma solução para amenizar a sobrecarga, porém não mencionada pelo autor, que aborda um outro tipo de tecnologia chamada de *Chronos*. O trabalho de Kyoung-Don *et al.* (2007) foi encarado aqui simplesmente como um exemplo de que existe o problema de sobrecarga em servidores de banco de dados e é preciso realizar trabalhos que ajudem a diminuir esse problema.

Com relação à avaliação de desempenho de sistemas, o trabalho de Pires *et al.* (2006) tem a proposta de apresentar um estudo comparativo de desempenho entre os servidores de banco de dados MySQL e PostgreSQL, com a utilização da ferramenta *Open Source Database Benchmark* (OSDB), em uma plataforma GNU/Linux. O trabalho consiste em analisar os SGBDs sob as métricas geradas pelo OSDB e fomentar possíveis melhorias em suas funcionalidades.

¹ Informações de pesquisas recorrentes de usuários [Mitra 2001].

De outra forma, o trabalho de Silva (2006) tem o objetivo de avaliar o desempenho dos principais métodos de armazenamento de objetos multimídia, com uso de *benchmark*. E usa como estudo de caso para avaliação o SGBD PostgreSQL.

Devemos também ressaltar o trabalho de Aquino (2007), que apresenta o sistema VEPersonal (*Personalized Virtual Environment*), uma infraestrutura para gerenciamento de componentes de realidade virtual, o qual permite adaptar ambientes virtuais em função do nível de conhecimento ou das necessidades do usuário. E teve o objetivo de propor, construir e validar uma infraestrutura para fazer o gerenciamento de componentes de ambientes virtuais adaptativos.

No entanto, um dos problemas principais do trabalho é a construção de mundos virtuais utilizando objetos 3D armazenados em um SGBD. Contudo, o armazenamento e a recuperação de objetos em um SGBD podem ser realizados por linguagens de consulta com suporte a XML. E partir disso, surge à necessidade de comparar SGBDs com XML nativo com SGBDs com suporte a XML [Aquino 2007].

Enfim, é notável a relevância da investigação aqui proposta, já que visão é um recurso que possui uma capacidade de diminuir um problema muito comum em servidores de grande porte, que é a sobrecarga. E conforme mostrado em trabalhos anteriores, existe uma necessidade de analisar e comparar tecnologias diferentes sob certos aspectos funcionais. E que essas análises sejam feitas de forma automatizada e metodológica, para que haja uma relevância e coerência na decisão de escolha do *software* apropriado para cada caso.

3. Metodologia

A metodologia usada na avaliação dos SGBDs foi a GQM (*Goal - Question - Metric*). No qual a ideia básica é derivar métricas de *software* a partir de questões que queremos responder e objetivos que queremos alcançar [Guarizzo 2008]. Com o intuito de mostrar quais das métricas fornecidas pela ferramenta JMeter vão ser utilizadas neste trabalho.

A seguir são especificados os pontos GQM dessa metodologia:

- Objetivo (*Goal*): o objetivo desse trabalho é descobrir se as consultas com recurso de visão podem trazer um ganho de desempenho em tempo de resposta para servidores de banco de dados. E também investigar se consultas com visão podem ter desempenhos diferentes em SGBDs distintos.
- Questões (*Question*): Consultas com visão são melhores que consultas sem esse recurso? Consultas com visão podem trazer um ganho significativo para servidores de banco de dados sobrecarregados?
- Métricas (*Metric*): Com base nos objetivos e nas questões levantadas, percebe-se que as métricas **média** e **série 90%**², fornecidas pelo JMeter, são suficientes para gerar o conhecimento necessário para atingir os objetivos. Essas medidas serão usadas para compor as fórmulas abaixo.

² Essa métrica mostra que 90% das amostras atingiram um determinado tempo.

Média Harmônica: serve para achar o *throughput* por unidade de tempo de cada requisição. Ela é definida pela fórmula mostrada logo a seguir, onde n é o número de requisições, e t é o tempo de resposta para cada requisição [Silberchatz *et al.* 2006].

$$\text{Throughput} = n / ((1 / t_1) + (1 / t_2) + \dots + (1 / t_n))$$

Speedup: ajuda a mostrar o ganho da diferença de desempenho entre as consultas, ou seja, quanto maior o *Speedup*, maior será o ganho de desempenho. É dado pela fórmula a seguir:

$$\text{Speedup} = \text{oldRequestTime} / \text{newRequestTime}$$

4.1. Descrição dos Testes

Todos os testes foram realizados em um notebook DELL® Inspiron 1545, com as seguintes configurações: processador Intel® Pentium Dual-Core CPU T4200 @ 2.00GHz, 2000Mhz, 2 núcleos, 2 processadores lógicos, com 512KB de *cache*, RAM de 3GB e disco SATA de 150 GB e 5300 rpm.

A etapa inicial compreende na instalação dos servidores de banco de dados MySQL 5.5.29 e o PostgreSQL 8.3. Os dois servidores foram instalados no sistema operacional Windows, e como o sistema operacional não influencia no resultado dos testes, esse foi escolhido meramente por proporcionar comodidade ao pesquisador.

A escolha dos SGBDs se deve ao fato de serem os mais utilizados pelos usuários, o que é refletido nas várias pesquisas feitas sobre eles, mostradas inclusive nos trabalhos relacionados. E também por termos permissão para realizar testes e divulgá-los, o que não seria possível em servidores de banco de dados proprietários, em que por questões de *marketing*, não permitem a divulgação dos resultados [Pires *et al.* 2006].

Para realizar os testes foi usada uma base de dados de gerenciamento escolar, já igualmente predefinida para os dois SGBDs, que contém 103 tabelas e 97 relacionamentos, e com um tamanho de aproximadamente 135MB. Em seguida foram construídas 4 consultas para os dois servidores, com 5, 10, 15 e 20 relacionamentos cada.

4.2. Carga de Trabalho

Pretende-se com as cargas de trabalho, esforçar ao máximo o servidor, tentando simular uma condição de uso real e observar o seu comportamento. Elas foram impostas aos dois SGBDs testados em 4 fases, como seguem abaixo:

- 1º Fase: Foram executadas as 4 consultas sem o recurso de visão (CTs) com 4, 8, 16, 32, 64, 128 usuários simultâneos de cada vez, no MySQL;
- 2º Fase: No PostgreSQL foram executadas as 4 (CTs) com a mesma quantidade de usuários simultâneos por vez;
- 3º Fase: Depois, foram executadas no MySQL as 4 consultas com o recurso de visão (CVs) também com 4, 8, 16, 32, 64, 128 usuários simultâneos;
- 4º Fase: E por fim, executamos no PostgreSQL as 4 (CVs), também com a mesma quantidade de usuários simultâneos descrita acima.

É importante enfatizar que tanto as (CTs) e as (CVs) construídas nos dois SGBDs, não usam o recurso de índices. Os índices são usados pelos servidores de banco de dados para obter um acesso aleatório rápido às tuplas do banco, onde cada índice está associado a uma determinada chave de busca [Silberchatz *et al.* 2006].

4. Resultados Obtidos

Para realizar os testes, alguns cuidados foram tomados para que não fossem considerados certos dados que não seguem um padrão em relação aos demais. Para isso foi usado a métrica série 90% (fornecida pelo JMeter), e comparou-se com a média dos tempos de resposta das requisições. Só foram validadas as médias que ficaram entre um intervalo de 10% (para mais ou para menos), do resultado gerado pela série 90%. Então se afirma que no mínimo 80% dos tempos de resposta estavam bem próximos à média.

Após isso, já com todos os resultados em mãos, foi utilizado a Média Harmônica para encontrar o *throughput* para cada tempo de resposta retornado. E para saber qual servidor de banco de dados teve um maior desempenho nas consultas com visão é usada a fórmula de *Speedup*. Com ela é possível quantificar quanto de ganho se obteve em relação aos dois tipos de consulta.

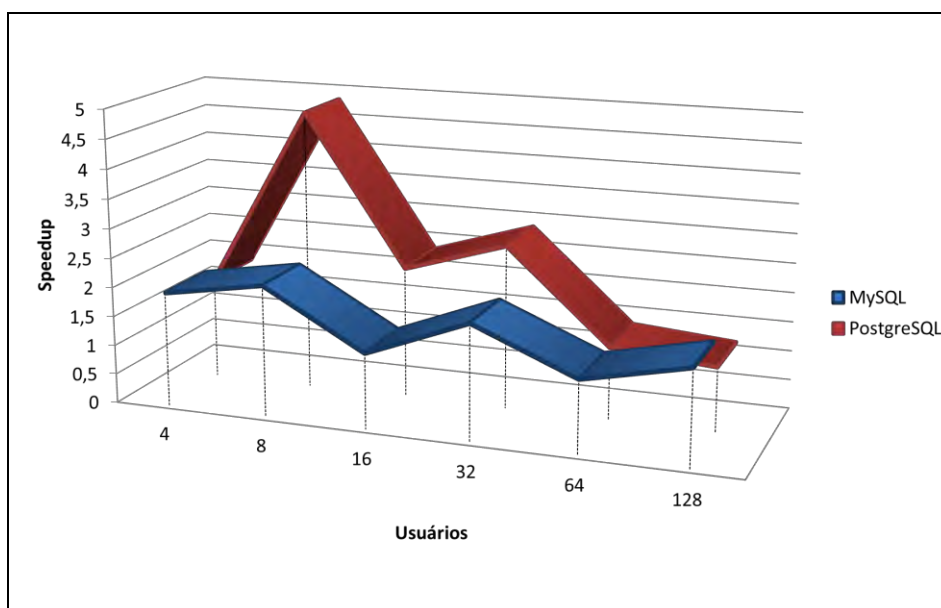


Figura 1. Comparação de *Speedup* das consultas nos SGBDs.

Na Figura 1 é mostrado um comparativo do *Speedup* entre o MySQL e o PostgreSQL, onde foi feita uma média do ganho das consultas com os diferentes relacionamentos. É possível notar que o PostgreSQL se mostrou com um resultado melhor que seu concorrente.

Por fim, é mostrada na Figura 2 a diferença total de ganho entre os dois SGBDs. O MySQL obteve um ganho de aproximadamente 70%, e o PostgreSQL obteve um ganho de 130%. Portanto, conclui-se que o PostgreSQL suporta melhor o recurso de visão, e tem uma diferença significativa de desempenho comparado com o MySQL.

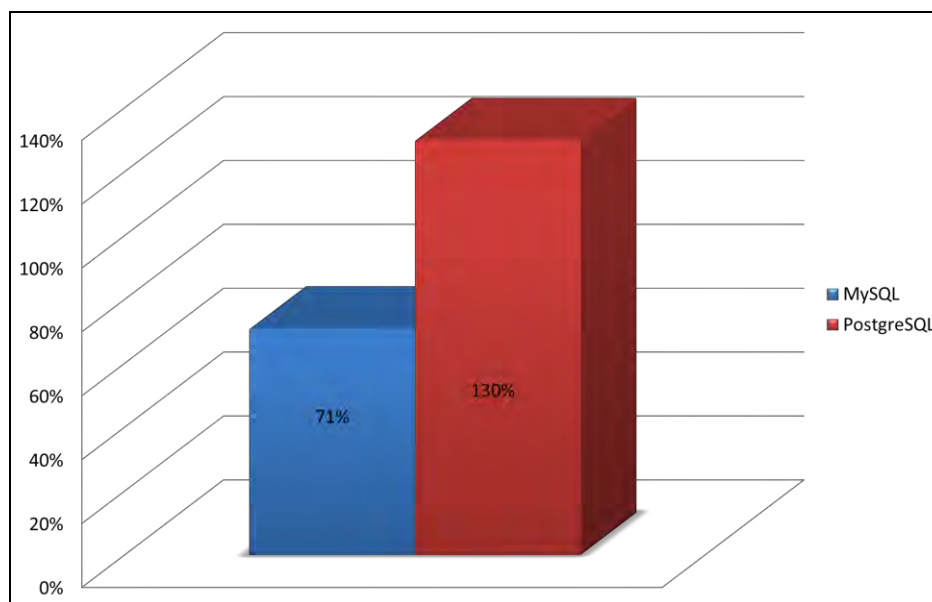


Figura 2. Desempenho das visões nos SGBDs.

5. Conclusão e Trabalhos Futuros

A sobrecarga de consultas em banco de dados é um problema que tende a se agravar com o passar do tempo. A enorme quantidade de informações requisitadas por dia pode ultrapassar a casa dos trilhões³. Desta forma, é necessário o uso de técnicas que minimizem esse intervalo de tempo, entre requisição e resposta, e alivie o meio de comunicação utilizado.

Baseado nisso, tem-se visto que o recurso de visão conseguiu retornar valores expressivos para redução de tempo de resposta, e podendo assim, contribuir com uma diminuição significativa da sobrecarga de servidores de dados. Além disso, foi possível observar que existem diferenças de desempenho de funcionalidades específicas entre SGBDs distintos, e a investigação será de grande ajuda para quem está em busca de uma ferramenta para um tipo de aplicação que se preocupa na maioria do tempo a responder requisições.

Como trabalhos futuros, é pretendido realizar esses mesmos testes sob consultas em diferentes SGBDs, mas agora usando visões materializadas, onde é guardado o resultado da consulta em vez do resultado do gerador de código, no qual pode-se observar se elas possuem diferenças de desempenho das consultas com recurso de visão virtual, a qual foi abordada neste trabalho. E verificar também como seria o seu comportamento em meio às atualizações em tempo real.

Também é desejável, aprofundar mais nas atividades exercidas pelo SGBD para responder a uma requisição, principalmente investigar o funcionamento do otimizador da consulta, e ver qual é a dificuldade de se chegar a um caminho ótimo em um tempo razoável, com o objetivo de verificar qual seria o motivo de diferentes SGBDs terem um tempo de resposta variável para consultas idênticas.

³ <http://www.realtimestatistics.org/>

References

- Aquino, Marcus S. (2007) “VEPersonal - Uma Infra-Estrutura para Geração e Manutenção de Ambientes Virtuais Adaptativos”. Tese de Doutorado, Universidade Federal de Pernambuco, Recife, PE.
- Ciferri, Ricardo R. (1995) “Um Benchmark Voltado à Análise de Desempenho de Sistemas de Informações Geográficas”. Dissertação de Mestrado, Universidade Estadual de Campinas, Campinas, SP.
- Date, C. J. (2003), Introdução a Sistemas de Banco de Dados, Editora Elsevier, 4ª edição.
- Elmasri, Ramez; Navathe, Shamkant B. (2005), Sistemas de Banco de Dados, Editora Pearson Addison Wesley, 4ª edição.
- Felea, Victor; Balta, Marian. (2007) “On Equivalence Of Queries Using Views”, In: International Symposium on Symbolic and Numeric Algorithms for Scientific Computing, IEEE Conference Publications.
- Guarizzo, Karina. (2008) “Métricas de Software”. Monografia de Graduação, Faculdade de Jaguariúna, Jaguariúna, SP.
- Halevy, A. Y. (2001) “Answering Queries Using Views: A Survey”, In: Department of Computer Science and Engineering, University of Washington.
- Halili, Emily H. (2008), Apache JMeter: A Practical Beginner's Guide to Automated Testing and performance measurement for your websites. Editora Packt Publishing, 1ª edição.
- Kyoung-Don, K.; Sin, P. H.; Jisu, O.; Son, S. H. (2007) “A Real-Time Database Testbed and Performance Evaluation”, In: 3th IEEE International Conference, Embedded and Real-Time Computing Systems and Applications.
- Mitra, Prasenjit. (2001) “An Algorithm for Answering Queries Efficiently Using Views”, In: 12th Australasian Database Conference, IEEE Conference Publications.
- Pires, Carlos E. S.; Nascimento, Rilson O.; Salgado, Ana C. (2006) “Comparativo de Desempenho entre Bancos de Dados de Código Aberto”, In: Escola Regional de Banco de Dados, Anais da ERBD06, Porto Alegre.
- Ramakrishnan, Raghu; Gehrke, Johannes. (2008), Sistemas de Gerenciamento de Banco de Dados. Editora McGraw-Hill, 3ª edição.
- Santos, Ismaley S.; Neto, Pedro A. S. (2008) “Automação de Testes de Desempenho e Estresse com o JMeter”, In: II ERCEMAPI - Escola Regional de Computação, São Luiz.
- Silberchatz, Abraham; Korth, Henry F.; Sudarshan, S. (2006), Sistemas de Banco de Dados. Editora Elsevier, 5ª edição.
- Silva, Ricardo C. (2006) “Benchmark em Banco de Dados Multimídia: Análise de Desempenho em Recuperação de Objetos Multimídia”. Dissertação de Mestrado, Universidade Federal do Paraná, Curitiba.