

Um *Framework* para a Avaliação de Desempenho de Bancos de Dados

Lucas Lemos¹, Pedro Holanda¹, José Maria Monteiro¹

¹ Universidade Federal do Ceará - UFC

lucaslemons@lia.ufc.br, pedrothiago@lia.ufc.br, monteiro@lia.ufc.br

Abstract. *This paper presents a multi-platform, multi-benchmarks and multi-DBMS framework, called dBest (database benchmark test toolkit), that provides support for carrying out performance evaluations on databases. The dBest enables quick prototyping of new benchmarks and easy inclusion of new DBMSs. Experimental results indicate that the use of the proposed framework facilitates performance testing, reducing the development time through software reuse and reducing the time spent in various evaluation stages.*

Resumo. *Este trabalho apresenta um framework, denominado dBest (database benchmark test toolkit), multi-plataforma, multi-benchmark e multi-SGBD que fornece suporte para a realização de avaliações de desempenho em bancos de dados. O dBest possibilita a rápida prototipagem de novos benchmarks e a fácil inclusão de novos SGBDs. Os resultados dos experimentos realizados indicam que a utilização do framework proposto facilita a realização de testes de desempenho, diminuindo o tempo de desenvolvimento através do reuso de software e reduzindo o tempo gasto nas diversas etapas do processo de avaliação.*

1. Introdução

As aplicações de bancos de dados têm se tornado cada vez comuns, variadas e complexas. Atualmente, estas aplicações podem ser caracterizadas pelo grande volume de dados armazenados e pela elevada demanda por disponibilidade e desempenho. Sendo que o desempenho está relacionado ao grau de satisfação dos clientes em relação aos tempos de resposta dos comandos SQL submetidos ao Sistema de Banco de Dados (SBD). Contudo, diversos fatores afetam o desempenho de um SBD: há elementos externos, tais como equipamentos físicos (memória, processador, discos, etc.), Sistema Operacional e questões de conectividade. Também existem influências internas que variam desde configurações de parâmetros, níveis de concorrência, uma correta configuração de índices, dentre outras.

Neste contexto, frequentemente, nos deparamos com a necessidade de avaliar o desempenho de um SBD. Entretanto, não existe um meio universal com o qual possamos realizar esta avaliação. Com base nesta constatação surgiram os *benchmarks* de bancos de dados. *Benchmarks* são padrões utilizados em comparações de sistemas computacionais. A aplicação de testes seguindo tais padrões gera medidas quantitativas de desempenho capazes de serem comparadas com as de outros sistemas. Dentre os *benchmarks* para bancos

de dados destacam-se: o OSDB (*Open Source Database Benchmark*) [OSDL 2007] e os *benchmarks* mantidos pelo TPC (*Transaction Processing Performance Council*) [Council 2007], tais como o TPC-C, o TPC-H e o TPC-E.

Todavia, um *benchmark* é apenas um padrão para medida ou avaliação. Na prática, para se avaliar um SBD usando um determinado *benchmark* é necessário uma ferramenta de *software* a fim de realizar um conjunto restrito e pré-definido de operações (denominado carga de trabalho) e retornar um resultado (uma métrica) que descreva o comportamento do sistema. Mesmo com a ajuda dessas ferramentas, realizar uma avaliação de desempenho de banco de dados não é uma tarefa trivial e envolve etapas complexas e trabalhosas, tais como: criar o esquema do banco de dados; realizar a carga dos dados; executar a carga de trabalho de teste; além de coletar e armazenar estatísticas sobre o teste realizado.

Assim, podemos perceber que realizar testes de desempenho em diferentes SGBDs (Sistemas de Gerenciamento de Bancos de Dados) que podem executar em plataformas de *software* (Sistemas Operacionais) e *hardwares* distintos é uma tarefa bastante complexa. Atualmente existem algumas ferramentas (denominadas *toolkits*) que facilitam este processo. Contudo, essas ferramentas são construídas para SGBDs específicos e, em geral, funcionam apenas em um determinado Sistema Operacional.

Neste trabalho, projetamos e implementamos um *framework*, denominado dBest (*database benchmark test toolkit*), multi-plataforma, multi-*benchmark* e multi-SGBD que permite aos usuários realizar testes de desempenho sobre qualquer SGBD, em qualquer Sistema Operacional, utilizando diferentes *benchmarks*. O dBest fornece suporte para todas as etapas envolvidas no processo de avaliação de desempenho de bancos de dados, possibilita a rápida prototipagem de novos *benchmarks* e a fácil inclusão de novos SGBDs. Tudo isso baseado na edição de arquivos de configuração. A fim de comprovar os benefícios da utilização do *framework* proposto, implementamos o suporte para testes de desempenho utilizando dois *benchmarks* distintos (TPC-C e TPC-H) em três SGBDs diferentes (Oracle, SQL Server e PostgreSQL).

O restante deste artigo está organizado da seguinte forma: a seção 2 apresenta os trabalhos relacionados; a seção 3 discute os requisitos, a arquitetura e as funcionalidades do *framework* proposto e a seção 4 conclui este trabalho e aponta direções para futuras pesquisas.

2. Trabalhos Relacionados

Poucos trabalhos acadêmicos relacionados à criação e ao aperfeiçoamento de *benchmarks* têm sido publicados [Gray 1992, Seng et al. 2005, Zhao and Truhlar 2005]. Um dos principais motivos para isso é a restrição imposta pela “cláusula DeWitt”, que proíbe a publicação de testes de desempenho envolvendo bancos de dados comerciais sem a prévia autorização do fabricante. Por esta razão, vários *benchmarks* foram construídos para comparar o desempenho de SGBDs de código aberto, tais como OSDB [OSDL 2007] e TPCC-UVa [Llanos 2006].

Por outro lado, várias ferramentas (*toolkits*) voltadas para auxiliar o processo de avaliação de desempenho de bancos de dados têm sido propostas. A Tabela 1 apresenta um resumo dessas ferramentas ^{1 2}. Contudo, tais ferramentas apresentam, em geral, diversas limitações: são construídas para SGBDs específicos, funcionam apenas em um determinado Sistema Operacional, fornecem suporte para poucos *benchmarks*, não fornecem suporte para o armazenamento dos resultados dos testes e não auxiliam na interpretação desses resultados.

Ferramenta	Open Source	Tipo de Teste	SGBDs	SOs	Métricas
OSDL DBT	Sim	Baseado no TPC	PostgreSQL	Linux	Vazão, CPU, I/O, Memória
PolePosition	Sim	Mapeamento O-R	HSQLDB, db4o, Versant	Todos	-
jTPCC	Sim	Baseado no TPC-C	Oracle, MySQL e PostgreSQL	Todos	-
BenchmarkSQL	Sim	Baseado no TPC-C	Oracle, SQL Srv e PostgreSQL (*)	Todos	-
Bristlecone	Sim	SELECT + INSERT / UPDATE / DELETE	-	Todos	Vazão e Tempo de Resposta
Swingbench	Sim	OrderEntry, SalesHistory, CallingCircle	Oracle	Todos	Vazão e Tempo de Resposta
TPCC-UVa	Sim	Baseado no TPC-C	PostgreSQL	Linux	Vazão
Oracle B. Kits	Não	-	Oracle	-	-
dBest	Sim	Baseado no TPC + Teste Customizado	Oracle, SQL Srv e PostgreSQL (*)	Todos	Vazão e Tempo de Resposta

Tabela 1. Análise Comparativa das Principais Ferramentas.

3. Um Framework para a Avaliação de Desempenho de Bancos de Dados

Nesta seção serão apresentados os requisitos que nortearam a concepção do *framework* proposto, além de sua arquitetura e principais funcionalidades.

3.1. Requisitos

Após uma análise detalhada das ferramentas existentes identificamos um conjunto de características essenciais para um *software* de avaliação de desempenho em bancos de dados. Essas características são discutidas a seguir:

¹* Indica a utilização de SGBDs com suporte ao padrão SQL ANSI.

²- Indica informação não disponibilizada ou não encontrada.

- **Representatividade:** Suporte à maior quantidade possível de *benchmarks*.
- **Portabilidade:** Possibilidade de executar em diferentes plataformas e SOs.
- **Repetibilidade:** Quando uma avaliação (teste) é aplicada no mesmo ambiente, mais de uma vez, ele deve produzir resultados semelhantes.
- **Elasticidade:** Possibilitar avaliações em ambientes com diferentes capacidades.
- **Não Intrusividade:** A ferramenta não deve envolver alterações no código fonte, metabase ou estrutura dos SGBDs utilizados.
- **Simplicidade:** A ferramenta deve ser de fácil utilização pelos usuários e deve ser possível adicionar novos *benchmarks* e SGBDs de maneira rápida e simples.

Além disso, as seguintes funcionalidades são consideradas fundamentais:

- Possibilitar a criação do banco de dados, tabelas e índices necessários;
- Executar a carga dos dados (“povoar” as tabelas) (*restore* e *dump*);
- Acompanhar a execução da carga de teste;
- Coletar e armazenar estatísticas sobre a carga de trabalho em execução, tais como vazão e tempo de resposta das transações;
- Gerar relatórios e gráficos para auxiliar a análise dos resultados;

3.2. Arquitetura e Principais Funcionalidade

O núcleo do dBest consiste de doze classes. As classes que compõem o dBest são organizadas em quatro pacotes: *test*, *db*, *util* e *gui*. O pacote *test* contém as classes usadas na máquina de execução dos testes e que lidam com a realização e monitoramento dos testes, além de possuir as abstrações para representar as transações (consultas), *benchmarks* e o modelo de comportamento do ambiente de teste. O pacote *db* contém as classes que estão diretamente relacionadas como o armazenamento dos resultados dos testes. O pacote *util* contém as classes utilitárias do *framework*, como por exemplo, uma classe para criar e destruir *threads*. O pacote *gui* contém as classes utilizadas na criação da interface gráfica com o usuário.

O dBest fornece suporte para todas as etapas envolvidas no processo de avaliação de desempenho de bancos de dados. Assim, inicialmente, o usuário configura os parâmetros necessários para a conexão com o banco de dados a ser avaliado. Esses dados podem ser fornecidos por meio da interface gráfica do dBest (Figura 1) ou editando-se um arquivo de configurações (propriedades). Para isso o usuário deve informar: o SGBD, o nome do banco de dados, o usuário do SGBD e sua respectiva senha. O dBest já suporta os SGBDs: Oracle, SQL Server e PostgreSQL. Contudo, se o usuário desejar utilizar um outro SGBD (MySQL, por exemplo) basta editar o arquivo de propriedades indicando: o nome do SGBD, o *driver* JDBC a ser utilizado, a localização do *driver* e a *string* (URL) de conexão. Em seguida, o usuário deve indicar qual *benchmark* deseja utilizar. O dBest implementa os *benchmarks* TPC-C e TPC-H. Porém, outros *benchmarks* (customizados) podem ser facilmente adicionados ao dBest. Para isso, basta ao usuário editar um arquivo de propriedades indicando: o nome do *benchmark*, um arquivo de configuração contendo os comandos DDL necessários para a criação do banco de dados (tabelas, índices, etc) e um

outro arquivo contendo a carga de trabalho do *benchmark* (cláusulas SQL ou transações). Caso o banco de dados a ser avaliado ainda não tenha sido criado o dBest disponibiliza opções que permitem criar o esquema do banco de dados (via comandos DDL) e realizar a carga dos dados, se um arquivo de *backup* for fornecido.

Estando a conexão com o banco de dados configurada e o *benchmark* selecionado, o usuário pode especificar o tipo de teste que deseja realizar (Figura 1). Neste sentido, o usuário deve indicar:

- O tipo de teste, ou seja, se a duração do teste será definida por tempo ou por iteração. Se o teste for por tempo (por exemplo 30 minutos) a carga de trabalho do *benchmark* é continuamente executada até que se complete o tempo estipulado. No teste por iteração (por exemplo, 3 execuções) a carga de trabalho do *benchmark* selecionado é executada por completo uma vez para cada uma das iterações;
- A ordem do teste, que pode ser uma das três seguintes alternativas: sequencial, aleatória fixa e aleatória.

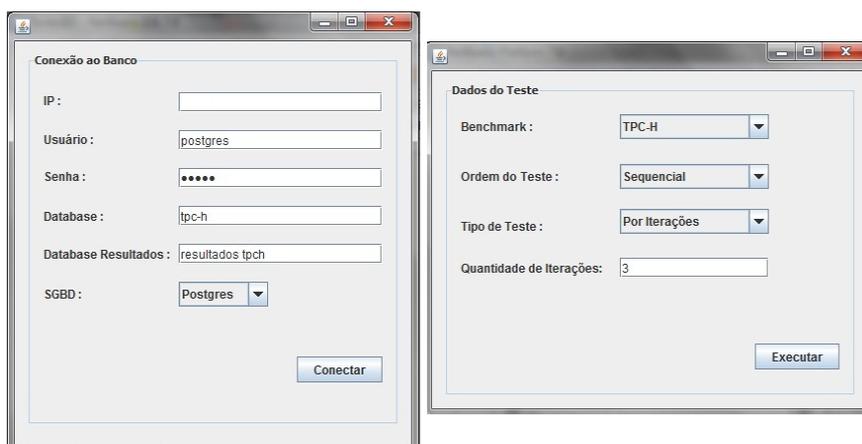


Figura 1. Telas de Configuração e de Seleção do Tipo de Teste.

Após o início de uma sessão de teste o dBest possibilita acompanhar e monitorar a execução do teste. Neste sentido o dBest informa por meio de sua interface gráfica: a) se a duração do teste for por tempo: o tempo decorrido, o tempo restante, as consultas já executadas juntamente com o tempo de execução e as consultas a serem executadas; b) se a duração for definida por iteração: o tempo decorrido, a quantidade de iterações concluídas, a duração das iterações concluídas, a quantidade de iterações restantes, as consultas já executadas juntamente com o tempo de execução e as consultas a serem executadas.

Ao final de uma sessão de testes o dBest armazena as estatísticas referentes ao teste realizado em um banco de dados. Adicionalmente, o dBest auxilia na interpretação dos resultados por meio de gráficos e relatório, os quais apresentam além das informações já fornecidas durante o monitoramento: o tempo de execução médio para cada consulta, a evolução dos tempos de execução para cada consulta e a vazão, como ilustra a Figura 2.

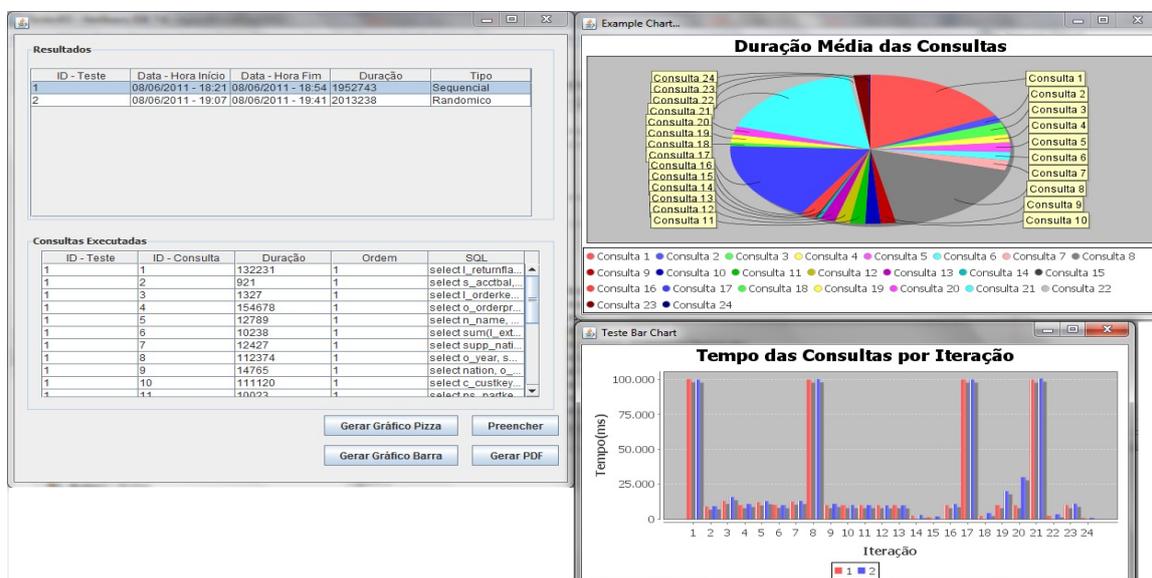


Figura 2. Tela de Resultados.

4. Conclusões e Trabalhos Futuros

Neste trabalho, foi apresentado um *framework* multi-plataforma, multi-*benchmark* e multi-SGBD, denominado dBest, o qual apóia a realização de testes de desempenho em bancos de dados. Como trabalhos futuros pretende-se realizar uma avaliação da usabilidade do dBest e adicionar o suporte necessário para a execução de testes de *stress*.

Referências

- Council, T. P. P. (2007). Tpc. Disponível em <http://www.tpc.org>.
- Gray, J. (1992). *Benchmark Handbook: For Database and Transaction Processing Systems*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.
- Llanos, D. R. (2006). Tpc-c-ua: An open-source tpc-c implementation for global performance measurement of computer systems. *SIGMOD Record*. ISSN 0163-5808.
- OSDL (2007). Osd. Disponível em <http://osdl.dbt.sourceforge.net/>.
- Seng, J.-L., Yao, S. B., and Hevner, A. R. (2005). Requirements-driven database systems benchmark method. *Decis. Support Syst.*, 38:629–648.
- Zhao, Y. and Truhlar, D. G. (2005). Benchmark databases for nonbonded interactions and their use to test density functional theory. *Journal of Chemical Theory and Computation*, 1:415–432.