

COMPARAÇÃO DOS SISTEMAS GERENCIADORES DE BANCOS DE DADOS ORACLE E POSTGRESQL COM O USO DA FERRAMENTA BENCHMARKSQL¹

POLLYANNA ESPINOSA MAIELLO²

Resumo: Juntamente com esse aumento, os bancos de dados passaram a ter um papel de destaque no processo de armazenamento das informações a fim de garantir uma maior segurança, estabilidade e acessibilidade a essas. Desta forma, a escolha de um Sistema de Gerenciamento de Banco de Dados que atenda as demandas dessas informações e que garantam um ganho de desempenho no gerenciamento dessas, pode significar uma vantagem competitiva. Nesse trabalho é apresentado um breve levantamento na literatura sobre a teoria das propriedades ACID dos bancos de dados e uma comparação de dois dos principais Sistemas Gerenciadores de Banco de dados disponíveis no mercado, utilizando para essa comparação o software BenchmarkSQL que simula o processo de análise Benchmark TPC-C que tem como objetivo gerar uma série de transações para medições posteriores.

Palavras-chave: Informática. Banco de Dados. SGBD. Benchmark

Abstract: Along with this increase, the databases have been given a prominent role in the storage process the information in order to ensure greater security, stability and accessibility to these. Thus, the choice of a database management system that meets the demands of the information and to ensure a performance gain in managing these can mean a competitive advantage. In this paper, we present a brief survey of the literature on the theory of ACID properties of databases, and a comparison of two of the main Systems Database Managers available in the market, using this comparison software BenchmarkSQL that simulates the scanning process Benchmark TPC-C which aims to generate a series of transactions for later measurements.

Keywords: IT. Database. DBMS. Benchmark

¹ Trabalho orientado pelo Prof. Ms. Marcelo Teixeira Torres

² maiello.pollyanna@yahoo.com.br
Universidade de Araraquara (UNIARA)

1 INTRODUÇÃO

De maneira geral um Sistema de Gerenciamento de Banco de Dados (SGBD) pode ser definido como uma série de programas e instruções correlacionadas para assegurar acesso a esses dados. O objetivo principal de um SGBD é fornecer formas de recuperar informações de um banco de dados, que pode ser definido como uma coleção de dados que contém informações relevantes a um negócio e/ou empresa, de forma conveniente e eficiente (SILBERSCHATZ; KORTH; SUDARSHAN, 2006).

Como o volume de informação armazenada por esses bancos de dados cresce exponencialmente à medida que o tempo passa, torna-se necessário buscar novas soluções para garantir maior segurança, estabilidade e acessibilidade a essas informações armazenadas. Ao escolher um SGBD para gerenciar as informações de sua empresa, o administrador do banco de dados precisa decidir qual é o mais indicado entre os diversos SGBD's disponíveis no mercado de acordo com as características e categorias que atende melhor a sua necessidade.

A proposta desse trabalho foca, nesse sentido, na comparação de desempenho de dois SGBD's disponíveis no mercado, Oracle e PostgreSQL, analisando e detalhando suas principais características, com ênfase nas propriedades ACID³ através de uma revisão bibliográfica, afim de determinar as principais vantagens de cada um desses SGBD's e assim auxiliar na tomada de decisão no que tange a escolha de um SGBD para cada tipo de negócio específico.

A seção introdutória é seguida por outras quatro seções. A seção seguinte oferece uma breve apresentação acerca dos SGBD's estudados nesse trabalho. Na seção 3 são descritas as propriedades ACID, seus princípios de funcionamento e sua importância para o correto funcionamento de um SGBD. Na seção 4 é apresentado o estudo de caso comparativo entre os dois SGBD's. Na última seção é exibida a conclusão do trabalho e sugestões de trabalhos futuros.

2 PROPRIEDADES ACID

Um dos principais e mais importantes conceitos da teoria de banco de dados é a coleção de propriedades denominadas ACID. A sigla ACID faz referência ao acrônimo de Atomicidade, Consistência, Isolamento e Durabilidade. A confiabilidade de uma SGBD passa pela capacidade de garantir a apresentação sem falhas dessas quatro características (ELMASRI; NAVATHE, 2005).

Essas propriedades têm como principal função garantir que após a execução de uma transação pelo SGBD esse continue consistente e íntegro. Assim, é de suma importância estudar a atitude das propriedades ACID na atitude das transações no cenário da computação (ELMASRI; NAVATHE, 2005).

³ Acrônimo utilizado para representar as propriedades Atomicidade, Consistência, Isolamento e Durabilidade.

2.1 Atomicidade

Atomicidade pode ser definida como a propriedade que trata das transações de forma atômica, ou seja, todas as instruções dentro de uma transação são tratadas de forma única e indivisível. A propriedade da atomicidade implementa o conceito de "tudo ou nada" em um SGBD, ou seja, todas as modificações nos dados presentes em uma transação são executadas com sucesso, ou nenhuma delas deverá ser executada (ELMASRI; NAVATHE, 2005).

É imperativo que cada SGBD implemente e mantenha a natureza atômica dos dados nas transações a respeito de qualquer falha do sistema operacional, do hardware ou mesmo do SGBD, sem que isso afete a premissa básica dessa característica (ELMASRI; NAVATHE, 2005).

2.2 Consistência

Consistência é a propriedade que assegura que todas as modificações contidas na transação serão efetuadas no banco de dados se esses forem consistentes. Assim, se todos os dados forem válidos a transação é efetuada, caso contrário, toda a transação deverá ser cancelada (ELMASRI; NAVATHE, 2005).

Essa propriedade implica que apenas dados válidos deverão ser escritos na base de dados, e caso alguma operação viole alguma regra de consistência a transação inteira deverá ser desfeita (*rolled back*) e a base de dados será restaurada a um último estado consistente de acordo com as regras estabelecidas, de outra forma, se a transação for bem-sucedida, a base de dados deverá passar para um novo estado válido, também consistente com as regras de negócio (ELMASRI; NAVATHE, 2005).

2.3 Isolamento

A propriedade de isolamento garante que os dados e as ações de uma transação não serão visíveis a nenhuma outra transação até que essa seja comprometida. Isso significa que não existirá relação de interdependência na execução de transações (PERKUSICH; TURNELL; PERKUSICH, 2000).

A propriedade requer que múltiplas transações ocorram ao mesmo tempo sem que uma cause algum impacto negativo em outras transações em execução. A propriedade de isolamento não define quem executará primeiro uma ação e sim que em uma situação de múltiplas transações, uma não interfira nas ações da outra (ELMASRI; NAVATHE, 2005).

Sendo (LIMA, 2008), isolamento ou o nível de isolamento pode ser definido de acordo com os termos de fenômenos que podem acontecer em transações concorrentes, conforme a tabela 1 abaixo:

Nível de isolamento	F0 (<i>Dirty Write</i>)	F1 (<i>Dirty read</i>)	F2 (<i>Non-repeatable Read</i>)	F3 (<i>Phantom</i>)
0	Impossível	Possível	Possível	Possível
1	Impossível	Impossível	Possível	Possível
2	Impossível	Impossível	Impossível	Possível
3	Impossível	Impossível	Impossível	Impossível

Tabela 1 – Níveis de Isolamento

Para (LIMA, 2008), o nível de isolamento define quais fenômenos podem acontecer em determinados níveis. Esses níveis de isolamento podem ser definidos como:

- O nível de isolamento F0 (*Dirty Write*) define que se uma transação T1 lê e modifica um elemento e outra transação T2 modifica e apaga o mesmo elemento e realiza um *commit*, se T1 tenta ler novamente esse elemento, ela pode encontrar o valor alterado por T2 ou descobrir que esse elemento não existe mais;
- O nível F1 (*Dirty Read*). Uma transação T1 modifica um elemento. Outra transação T2 lê o elemento antes de T1 concretizar o *commit*, se T1 realizar um *rollback*, T2 terá lido um valor que nunca foi gravado e, portanto, nunca existiu;
- O nível F2 (*Non-repeatable Read*). Uma transação T1 lê um elemento, uma transação T2 lê e modifica ou apaga o mesmo elemento e concretiza um *commit*, Se T1 tentar reler esse elemento, poderá receber um valor modificado ou descobrir que o elemento foi removido;
- O nível F3 (*Phantom*). Uma transação T1 lê um conjunto de valores N retornados a partir uma busca com um determinado critério, Uma transação T2 executa uma lista de comandos que satisfaça as mesmas condições de busca utilizadas por T1. Se a leitura inicial for refeita com os mesmos critérios de busca por T1, essa receberá um conjunto de valores diferentes.

A tabela 1 mostra uma matriz relacionando os níveis de isolamento com os fenômenos que podem acontecer em cada nível, por exemplo, no nível de isolamento 2 o único fenômeno que pode ocorrer é o *Phantom* (LIMA, 2008).

2.4 Durabilidade

A propriedade de durabilidade de um SGBD garante que as ações de uma transação com o banco não serão perdidas e são permanentes. Ou seja, uma vez que uma transação seja concretizada implica que essa não poderá ser perdida (PERKUSICH; TURNELL; PERKUSICH, 2000).

Assim, a durabilidade assegura que qualquer transação efetuada com sucesso, não deverá ser perdida. A propriedade da durabilidade é garantida através de uso de backups na base de dados e da aplicação de logs de transações que facilitem a restauração de transações que

forem bem-sucedidas, a despeito de qualquer falha de ambientes, como softwares e hardwares (ELMASRI; NAVATHE, 2005).

3 SGBD'S ANALISADOS

3.1 PostgreSQL

O PostgreSQL ocupa posição de destaque entre os SGBD's de código aberto disponível no mercado. Sendo até, considerado por muitos como o melhor SGBD de código aberto disponível atualmente. O PostgreSQL foi desenvolvido por várias pessoas, dentre as quais estavam um grupo de pesquisadores da universidade da UC Berkeley, sobre um projeto intitulado como *Ingres*. Esse projeto tinha como seu principal colaborador Michael Stonebreaker, sendo inclusive, considerado por muitos, como o "pai" do mesmo (OBER; HSU, 2011).

O PostgreSQL foi escrito em C e utiliza a extensão PL/pgSQL (Procedural Language/PostgreSQL Structured Query Language) do SQL. Esta extensão é baseada na extensão PL/SQL (Procedural Language/SQL) utilizada pelo Oracle, que por sua vez é baseada na Ada e tem como objetivo permitir a realização de operações computacionais mais complexas que o SQL, sendo ao mesmo tempo mais fácil de usar. Isso quer dizer que uma função criada nessa linguagem pode ser executada como uma ação realizada por uma *trigger* ou mesmo numa operação SQL (SAMPAIO; GOUVEIA; MARQUES, 2012).

O PostgreSQL utiliza uma sistema de *locking* denominado *Multiversion Concurrency Control*, ou MVCC. O PostgreSQL foi elaborado inicialmente para ser rodado nas plataformas Unix / Linux (TRAMONTINA, 2006).

O PostgreSQL não se limita as funcionalidades de um SGBD comum, ele é um SGBD Objeto-relacional. Isso quer dizer que, ele possui capacidade de armazenamento de dados mais complexos em suas colunas relacionais, além de poder armazenar informações no formato de objeto, formato esse utilizado pelas linguagens de programação que utilizam o paradigma de Orientação a Objeto. O PostgreSQL possibilita ainda que o usuário defina novos operadores e crie novos tipos de dados (OBER; HSU, 2011).

O PostgreSQL oferece suporte aos mais populares sistemas operacionais do mercado como Windows, Linux, MAC OS X, Sun Solares entre outros e pode ser instalado em um computador pessoal com 8MB de memória RAM e no mínimo 100MB de espaço de disco livre. Entre as principais características do PostgreSQL destacam-se o suporte a transações, suporte a triggers, stored procedures, não possui limite ao número de usuários, controle de integridade referencial, oferece controle de transações concorrentes e suporte nativo às quatro propriedade ACID (GROUP, 2005).

3.2 Oracle

O SGBD Oracle teve seu desenvolvimento iniciado a partir do final da década de 70 e atualmente está sendo distribuído na versão 12c. A principal característica que o Oracle oferece são ferramentas que buscam facilitar a vida do programador no processo de implementação de banco de dados (ORACLE, 2016).

O Oracle possui versões disponíveis para os sistemas operacionais Linux, Windows, AIX, Solaris, entre outros. O Oracle traz consigo uma grande gama de ferramentas que visam implementar facilidades para o desenvolvimento de *Data Warehouse's*, *clusters* de banco de dados, entre outros atributos que tem como objetivo aumentar a experiência do usuário com o (PRASS; PRASS, 2014). Segundo (OLIVEIRA, 2013) a sigla 'c' indica que esse é um SGBD e um banco de dados voltado para trabalhar nas nuvens (*cloud*).

Segundo (PRADO, 2012), entre as principais características apresentadas por esse SGBD, podemos citar a parte de segurança, como por exemplo:

- As ações do Oracle não são persistidas por padrão, ficando a cargo dos sistemas e/ou desenvolvedores executar essa ação, garantindo maior autonomia no manuseio dos dados;
- Leitura consistente dos dados, não permitindo que dois usuários vejam informações que ainda não tenha sido persistida pelo outro usuário, sem bloqueio de leitura e sem perigo de serem acessadas informações inconsistentes;
- Arquitetura flexível e com maior desempenho no acesso e gerenciamento das estruturas de memória do BD;
- Possui *Packages*, objetos que possibilitam agrupar e encapsular códigos de *stored procedures e functions*, entre outros

4 ESTUDO DE CASO

Para realização da comparação de desempenho dos SGBD's mencionados nesse trabalho foi utilizado o processo de Benchmark TPC-C, com a ferramenta BenchmarkSQL, simulando um ambiente próximo ao encontrado no dia a dia de uma empresa.

Uma vez com o ambiente montado, foi realizado uma simulação de carga de dados nos bancos de dados utilizando o software BenchmarkSQL e posterior execução de determinados conjuntos de transações por um determinado período de tempo. Os dois SGBD trabalharam com transações *On-Line Transaction Processing (OLTP)* em duas situações distintas, uma com uma quantidade razoável de transações.

Depois da execução das cargas de testes em ambos os SGBD's, coletou-se os dados obtidos dessas execuções e realizou-se uma análise comparativa entre os dois SGBD's observando o comportamento de cada um deles em relação à quantidade de transações realizadas.

4.1 Benchmark TPC-C

Benchmarks são ferramentas desenvolvidas com o intuito de avaliar o desempenho relativo de um objeto, e na maioria das aplicações, utiliza-se de uma série de testes padrões para isso. Quando falamos em Banco de Dados, a utilização de um *benchmark* é de suma importância para o entendimento de como um SGBD responde sob as mais variadas condições de utilização (MACHADO, 2010).

O Benchmark TPC-C é uma carga de trabalhos, composta por um conjunto de instruções de cinco transações de escrita e leitura que simula as diversas atividades encontradas em um ambiente OLTP de trabalho complexo. O objetivo do TPC-C é gerar dados relevantes referentes ao desempenho dos SGBD's, auxiliando ao usuário comparar de maneira objetiva esses sistemas de acordo com a sua capacidade de processamento de suas transações (MENDES, 2006).

As transações projetadas nesse *Benchmark* simulam um típico sistema de aquisição de produtos, com as principais funcionalidades encontradas nesse tipo de aplicação, tais como a criação de novos pedidos, registro de saída de produtos, confirmação de pedidos, realização de pagamentos, entre outras. Esse fornecedor atacadista representado por esse *Benchmark* possui um número determinado de distritos de vendas e lojas associadas, distribuídas por diversos lugares. Esses distritos e lojas aumentam à medida que a loja cresce, sendo que cada distrito tem a capacidade de atender 3.000 clientes. Os distritos são separados por armazém, onde cada armazém atende até 10 distritos. Todas as lojas possuem um estoque de 100K de produtos no seu portfólio (LIMA, 2008). A figura 1 detalha essa hierarquia entre os armazéns, distritos e clientes:

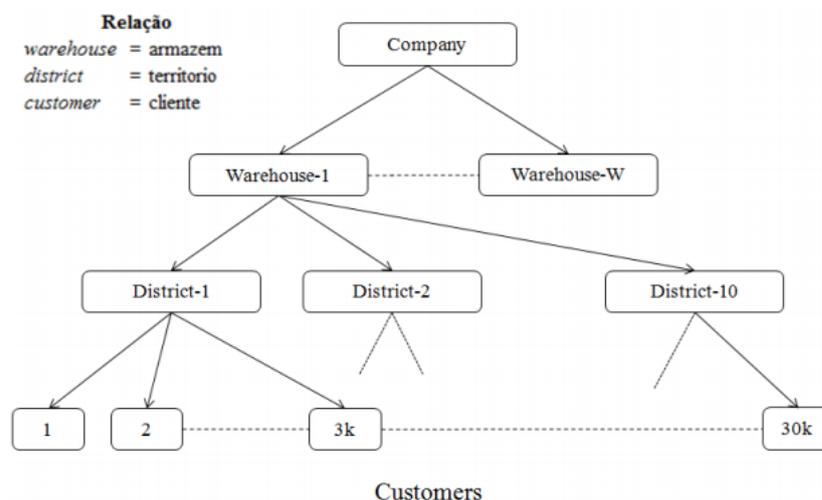


Figura 1 – Relação entre armazéns, distritos e clientes (LIMA, 2008)

O modelo TPC-C especifica um banco de dados composto por nove tabelas. A figura 2 abaixo mostra as tabelas desse banco e os seus respectivos relacionamentos:

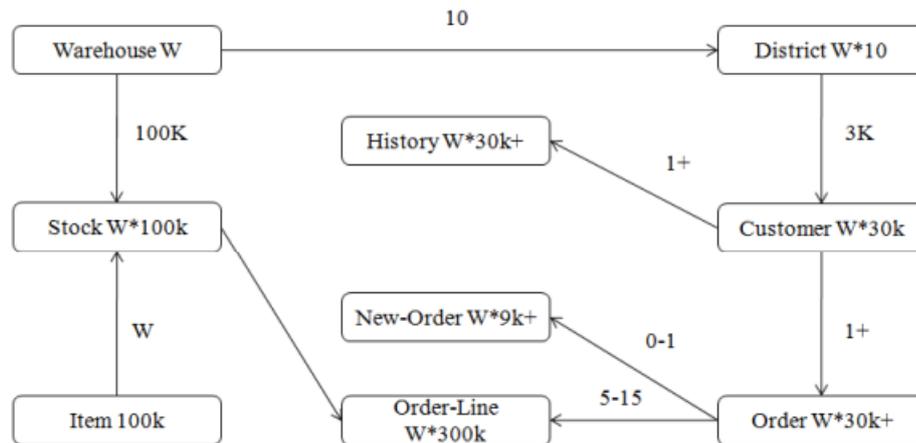


Figura 2 – Diagrama E-R do banco de dados TPC-C (LIMA, 2008)

Os números dentro de cada entidade mostram a cardinalidade das tabelas (quantidade de registros). Assim, a quantidade de cada entidade é proporcional ao número de linhas da tabela *warehouse* e os números próximos as linhas dos relacionamentos representam a cardinalidade dos relacionamentos (LIMA, 2008).

A metodologia do TPC-C define uma série de doze testes para identificar o nível de consistência estabelecida pela propriedade ACID do SGBD. Após o carregamento inicial dos dados, todos os doze testes devem apresentar resultados de que o banco de dados está íntegro e consistente (LIMA, 2008).

4.2 BenchmarkSQL

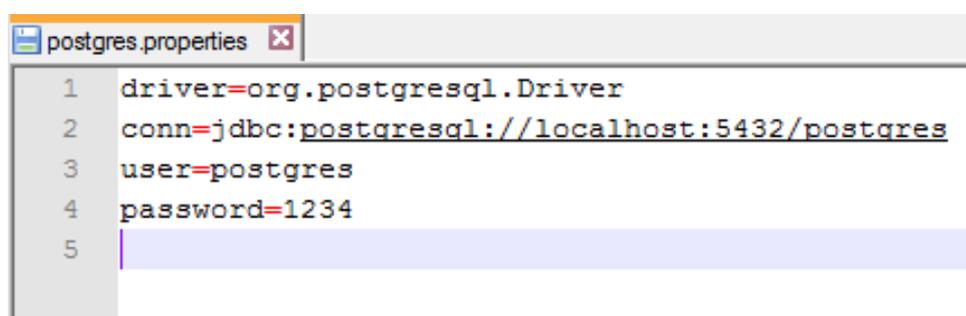
O BenchmarkSQL é uma ferramenta desenvolvida em linguagem de programação Java e que se assemelha bastante a especificação detalhada no Benchmark TPC-C. O diferencial dessa ferramenta é a utilização de *drivers* JDBC (*Java Database Connectivity*) para se comunicar com os SGBD's, permitindo assim que se comunique e seja compatível com uma ampla gama de SGBD's, como EnterpriseDB, PostgreSQL, Oracle, Sybase, SQLvr, MySQL, HsqlDB, JavaDB e FireBird (LIMA, 2008) (SUN, 2009).

A interface dessa ferramenta é bem intuitiva e similar as janelas encontradas nos sistemas Windows, possibilitando assim a execução de suas funcionalidades através de cliques dos botões (LIMA, 2008).

4.2.1 Configuração do BenchmarkSQL

O processo de configuração do BenchmarkSQL é composto por cinco etapas distintas a se saber: Configuração de comunicação com o SGBD, Criação das tabelas, Carga dos registros nas tabelas, Criação dos Índices e Execução do BenchmarkSQL (BENCHMARKSQL, 2006).

Configuração de comunicação com o SGBD: A configuração de acesso do BenchmarkSQL ao banco de dados é feita através de um arquivo de configuração específico para cada SGBD. Por exemplo, para a configuração do PostgreSQL é utilizado o arquivo: postgres.properties que contém o driver JDBC, caminho de acesso, usuário e senha, conforme figura 3:



```

1 driver=org.postgresql.Driver
2 conn=jdbc:postgresql://localhost:5432/postgres
3 user=postgres
4 password=1234
5

```

Figura 3 – Arquivo de Configuração do BenchmarkSQL - Fonte: próprio autor

- **Criação das Tabelas:** Nessa etapa é feita a carga das tabelas nos SGBD's supramencionados. Para tanto, foi executado o arquivo runSQL.bat para o sistema operacional Windows ou runSQL.sh para os sistemas baseados em Unix e mais os comandos:
runSQL PostgreSQL.properties sqlTableCreates
runSQL Oracle.properties sqlTableCreates
- **Carga das Tabelas:** Para executar essa ação, foi executado o arquivo loadData.bat para os sistemas Windows ou o loadData.sh para os sistemas operacionais baseados em Unix. Essa ação gerou uma base de dados aleatória baseada no método TPC-C com aproximadamente 600.000 registros para cada Warehouse informado.
- **Criação dos Índices:** Nessa etapa foram criados os índices e as chaves primárias que identificam as tabelas, sendo necessário para tanto a execução dos comandos:
runSQL PostgreSQL.properties sqlIndexCreates
runSQL Oracle.properties sqlIndexCreates
- **Execução do BenchmarkSQL:** após as etapas de configuração listadas anteriormente, o programa executando o arquivo runBenchmarkSQL.bat.

A figura 4 demonstra a tela inicial do programa BenchmarkSQL com as configurações iniciais de comunicação com o banco de dados PostgreSQL sendo exibidas.

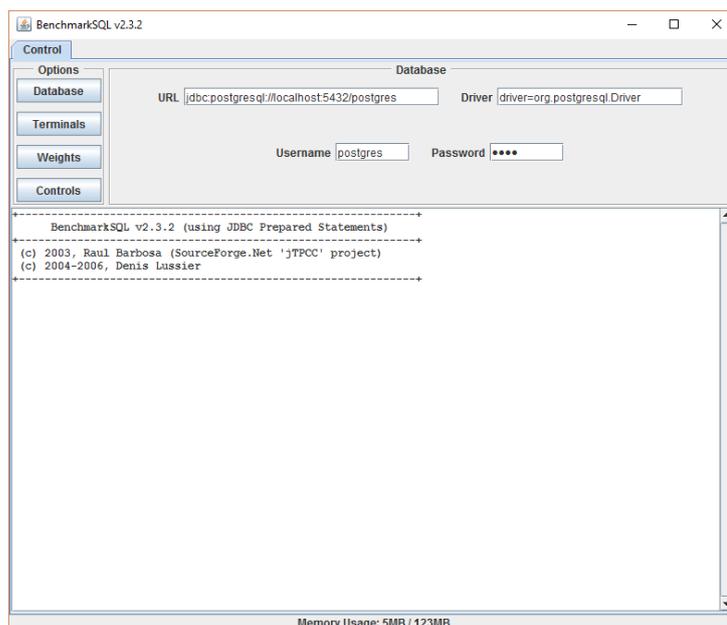


Figura 4 – Tela Inicial do BenchmarkSQL - Fonte: próprio autor

4.3 Ambiente de Testes

Para execução dos testes, foi utilizado um servidor Intel(R) Core (TM) i5-4210U de 2.40 GHz, com um disco rígido de 500GB, com um total de 8GB de memória RAM. O sistema operacional (SO) utilizado foi o Windows Server 2012 R2 projetado para arquitetura de 64 bits. Os SGBD's escolhidos para esse teste e instalados nessa máquina foram:

- PostgreSQL 9.5.0-1;
- Oracle XE 11G

Como o Benchmark utiliza-se da conexão JDBC para se comunicar com os SGBD's e os SGBD's escolhidos para esse teste suportam esse tipo de conexão, nenhuma outra configuração foi necessária no ambiente de testes e os SGBD's foram instalados em sua configuração padrão.

4.4 Simulação

Após a etapa de configuração e instalação dos SGBD's de testes, foi criada a simulação de desempenho, que constituiu na criação do ambiente que simulou uma quantidade razoável de transações OLTP durante um período de 10 minutos em cada um dos SGBD's, utilizando o software BenchmarkSQL como analisador de desempenho. Nessa etapa, foram utilizadas três relações entre Terminais x *Warehouse* diferentes, sendo essas configurações das relações definidas como:

- **Primeira Configuração:** 1 warehouse para 10 terminais;

- **Segunda Configuração:** 10 warehouse's para 50 terminais;
- **Terceira Configuração:** 20 warehouse's para 50 terminais.

Essa variação na quantidade de terminais em relação a quantidade de *Warehouse* visa aumentar a quantidade das transações realizadas pelos SGBD's e assim, aumentar a quantidade de processamento desses, afim de se testar esses sistemas em diferentes situações, simulando um ambiente empresarial em diferentes cenários de utilização.

O primeiro passo desse processo foi a criação das tabelas utilizadas nesse trabalho. Essa etapa inicial consiste na criação de um banco de dados chamado TEST, que por sua vez contém nove tabelas a qual será populada de acordo com a quantidade de terminais utilizado em cada simulação. No processo de geração e carga das tabelas foram geradas várias transações do tipo: *New-Order, Payment, Order-Status, Delivery e Stock-Leve*, foram efetuadas.

4.4.1 Critérios de Avaliação

Como forma de mensurar o desempenho dos SGBD's pelo BenchmarkSQL, utilizamos dois critérios de avaliações para definir as métricas de performance. A primeira métrica foi a quantidade de transações realizadas durante o período de tempo estipulado e a segunda métrica é o Tempo Médio das Transações TPC-C (TMTC), que nesse caso, será apresentada em milissegundos (ms).

Como forma de garantir uma maior precisão nos resultados, a cada execução as tabelas e os index foram deletadas, criados e carregados novamente no SGBD. Dessa forma, garantimos que a quantidade de registros não interfira no resultado final da simulação.

Assim, o próprio processo de escrita dos dados nas tabelas utilizadas pelo BenchmarkSQL serve como métrica de avaliação entre os SGBD's, uma vez que a quantidade e complexidade dos dados são praticamente os mesmos para ambos. Nesse quesito, a quantidade de linhas escritas por segundos e o tempo total gasto para finalizar o processo foram estimados.

4.4.2 Simulação 1

Primeiramente foi realizado uma análise de desempenho nos SGBD's utilizando a primeira configuração supramencionada. Para essa simulação foi carregado a quantidade de aproximadamente 600.000 tuplas de dados nas nove tabelas utilizadas pelo BenchmarkSQL. Nesse processo de carga inicial dos dados o Oracle teve um desempenho muito superior ao do PostgreSQL, terminando o processo em 17 segundos, contra 52 segundos do seu concorrente, executando 35.260 linhas e 11.525 linhas por segundo respectivamente.

Após esse processo de carga inicial foi rodado a etapa de processamento do Benchmark a qual obtivemos o seguinte resultado, conforme podemos observar na figura 5.

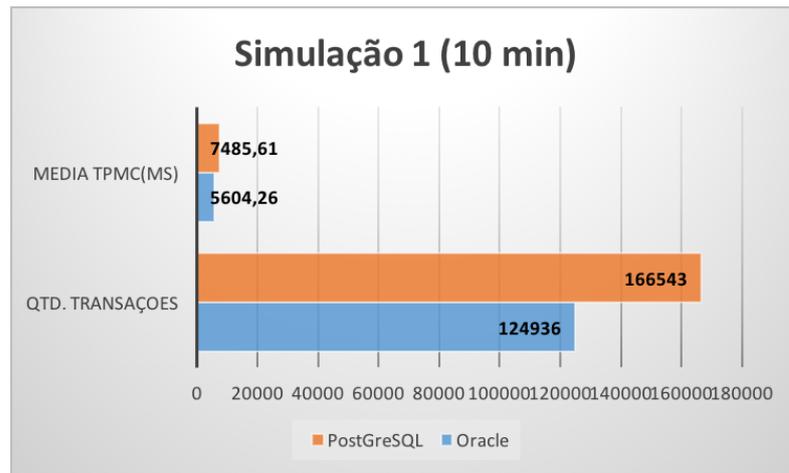


Figura 5 – Comparação de desempenho entre os SGBD's com a 1ª configuração

Observa-se que nessa primeira simulação, o SGBD Oracle teve um desempenho pior que o PostgreSQL, executando uma quantidade de 24,98% a menos transações que o executado pelo PostgreSQL.

4.4.3 Simulação 2

Nessa segunda simulação, foi utilizada a configuração 2 e para tal foram criadas as tabelas e feita a carga dos dados nessas. Nessa etapa, o processo para popular as tabelas com os dados levaram cerca de 190 segundos no Oracle contra 435 segundo no PostgreSQL, criando pouco mais de 5 milhões e 90 mil linhas em cada SGBD, sendo que o Oracle escreveu cerca 26.794 linhas por segundo, contra 11.701 linhas por segundo no PostgreSQL. O resultado dessa simulação podemos observar na figura 6.

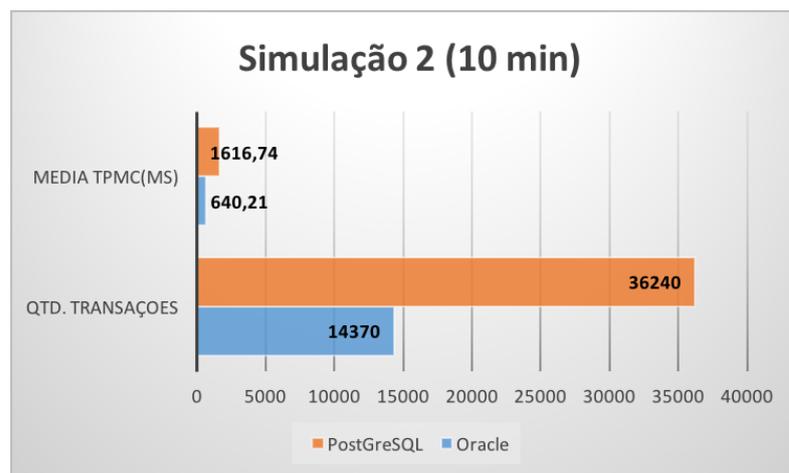


Figura 6 – Comparação de desempenho entre os SGBD's com a 2ª configuração

Podemos observar que nessa simulação o SGBD PostgreSQL se mostrou muito mais eficiente no processamento das transações do que o Oracle, executando cerca de 60,35% transações

a mais do que seu concorrente.

4.4.4 Simulação 3

Nessa simulação foram criadas cerca de 10 milhões e 80 mil linhas em cada SGBD para popular os 20 Warehouse's informados no exercício. Nesse processo, o Oracle se mostrou muito mais rápido no processo de criação das linhas, onde esse levou cerca de 318 segundos contra 747 segundos do PostgreSQL. Para tanto, o Oracle processou mais do que o dobro das linhas por segundo do que seu concorrente, sendo 31.700 linhas/segundo contra 13.490 linhas/segundo respectivamente.

Quanto ao resultado do processamento das transações no processo Benchmark podemos observar na figura 7.

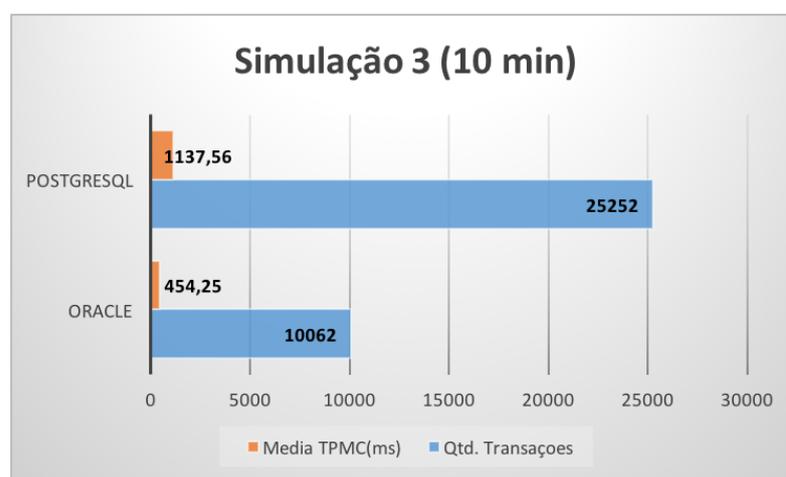


Figura 7 – Comparação de desempenho entre os SGBD's com a 3ª configuração

Assim como na simulação anterior, essa simulação com a terceira configuração apresentou um maior processamento de transações do PostgreSQL em relação ao Oracle. Nessa simulação, a diferença de processamento ficou na casa dos 60,15% de um SGBD para outro.

4.4.5 Análise dos Resultados

Conforme observado nas três simulações realizadas nesse trabalho, o SGBD Oracle se mostrou muito mais eficiente no processo de criação e carga dos registros da base de dados que foram utilizados como objeto de análise durante o processamento das transações.

Durante a execução do processo de *benchmark* por outro lado, o PostgreSQL mostrou-se muito mais eficiente, executando uma quantidade de transações muito superior ao número de transações executadas pelo Oracle.

Essa discrepância na quantidade de transações entre os dois SGBD's só aumentou com a quantidade de registros envolvidos no processo de *benchmark*. A medida que o número de

Warehouse aumentou, e conseqüentemente o número de registros utilizados nesse processo também, essa diferença ficou mais evidente, mostrando assim, que nesse tipo de teste, com os dois SGBD's instalados na sua configuração padrão e nesse tipo de ambiente detalhado no trabalho o PostgreSQL é a melhor opção para otimizar o processamento das transações.

5 CONCLUSÃO

Este artigo teve como objetivo demonstrar a importância dos SGBD's, especialmente nos dias atuais com a crescente demanda por armazenamento das informações pelas empresas e ter um SGBD que melhor atenda às suas necessidades, pode significar uma vantagem estratégica nesse mundo globalizado tão concorrido, e proporcionar uma análise em dois dos principais SGBD's do mercado a fim de facilitar o processo de tomada de decisão no processo de escolha do mesmo.

Foi mostrado algumas das principais características dos SGBD's e as diferenças entre eles. Também foram apresentadas as propriedades *ACID* utilizadas pela maioria dos SGBD's disponíveis no mercado e como essas propriedades contribuem para a melhor otimização dos dados em um ambiente computacional onde diversas transações concorrentes acontecem ao mesmo tempo.

Como forma de comparação entre os dois SGBD's foi utilizado o processo de *benchmark* TPC-C que simula o processamento transações a qual simula o ambiente real de uma empresa, utilizando-se para isso cinco operações de escrita e leitura nos SGBD's a fim de gerar uma massa de dados para futura análise e comparação entre esses.

A cada simulação, as tabelas eram eliminadas e recriadas novamente, assim como a carga dos registros foi refeita. Dessa forma, o processo de escrita dos registros básicos para as simulações a cada SGBD foi analisado a fim de estipular qual SGBD realiza esse processo de forma mais rápida. Nesse aspecto o Oracle se mostrou superior, escrevendo uma quantidade superior de linhas por segundo e finalizando o processo, nas três quantidades de dados, de forma muito mais rápida do que o PostgreSQL.

Já no processo de simulação das transações concorrentes efetuadas pela ferramenta BenchmarkSQL o PostgreSQL teve um desempenho superior ao Oracle nas três simulações realizadas, onde a discrepância entre o desempenho de um SGBD para outro só aumentou na proporção que os dados e concorrência das transações aumentavam.

Assim, para esse teste, podemos dizer que em um ambiente a qual o número de transação concorrente seja maior o PostgreSQL é mais aconselhado, uma vez que seu desempenho nos testes de *benchmark* foi melhor. Já em um ambiente onde não exista concorrências nas transações e o processo de escrita e leitura na base de dados não seja alto, o Oracle se mostrou superior, conforme demonstrado no processo de carga inicial dos dados.

Em trabalhos futuros pode ser explorado mais a fundo esse processo de análise dos dois SGBD's, mudando o tipo de ambiente e configuração da máquina no qual o teste foi realizado, bem como testar esses SGBD's com diferentes configurações, buscando utilizar ao máximo o seu potencial de processamento e propriedades que melhor se adequam a cada tipo de operação.

REFERÊNCIAS BIBLIOGRÁFICAS

BENCHMARKSQL. **Documentação do BenchmarSQL User Guide**. [S.l.], 2006. Disponível em: <<http://sourceforge.net/projects/benchmarksql/>>. Acesso em: junho de 2016.

ELMASRI, R.; NAVATHE, S. B. **Sistemas de banco de dados**. São Paulo: Addison Wesley Bra, 2005.

GROUP, T. P. G. D. **Documentação do PostgreSQL 9.5.2**. [S.l.], 2005. Disponível em: <<http://www.postgresql.org/docs/9.0/static/>>. Acesso em: maio de 2016.

LIMA, M. R. **Execução distribuída de benchmarks em Sistemas de bancos de dados relacionais**. Curitiba: Universidade Federal do Paraná, 2008.

MACHADO, C. A. **Estudo sobre a otimização de desempenho em banco deda dos PostgreSQL**. Lages: Universidade do Planalto Catarinense, 2010.

MENDES, M. R. N. **Análise de desempenho de sistemas OLTP utilizando o Benchmark TPC-C**. Recife: Universidade Federal de Pernambuco, 2006.

OBE R.; HSU, L. **PostGIS in action**. Brasil, 2011. Disponível em: <<http://dl.acm-.org/citation.cfm?id=2018871>>. Acesso em: 24/Setembro/2015.

OLIVEIRA, R. **O que vem por ai, Oracle 12c**. [S.l.], 2013. Disponível em: <<http://olivertconsultoria.blogspot.com.br/2013/02/o-que-vem-por-ai-oracle-12c.html>>. Acesso em: maio de 2016.

ORACLE, B. **A História do Oracle: Inovação, Liderança e Resultados**. [S.l.], 2016. Disponível em: <<http://www.oracle.com/br/corporate/newsroom/story-346137-ptb.html>>. Acesso em: maio de 2016.

PERKUSICH, M. L. B.; TURNELL, M. d. F.; PERKUSICH, A. **Modelagem de Banco de Dados em Tempo-real**. João Pessoa: Universidade Federal da Paraíba, 2000.

PRADO, F. **Qual é o melhor Banco de Dados: ORACLE ou SQL SERVER**. [S.l.], 2012. Disponível em: <<http://www.fabioprado.net/2012/01/qual-e-o-melhor-banco-de-dados-oracle.html>>. Acesso em: maio de 2016.

PRASS, F.; PRASS, F. **Oracle e Delphi uma visão geral do melhor dos bancos de dados**. [S.l.], 2014. Disponível em: <http://www.devmedia.com.br/articles/viewcomp.asp?comp=1311&hl=*oracle*>. Acesso em: maio de 2016.

SAMPAIO, B.; GOUVEIA, J.; MARQUES, P. **Relatório de comparação de SGBDs**. Lisboa-PT: Universidade Nova de Lisboa, 2012.

SILBERSCHATZ, A.; KORTH, H. F.; SUDARSHAN, S. **DBA RAC 11g Arquitetura: Instalação, Administração e Performance (Tradução da 5ª Edição)**. Rio de Janeiro: Editora Campus., 2006.

SUN. **Jdbc drivers**. [S.l.], 2009. Disponível em: <<http://developers.sun.com/Product/jdbc/drivers>>. Acesso em: maio de 2016.

TRAMONTINA, G. B. **Database Tuning: Configurando o Interbase e o PostgreSQL**. Campinas: Unicamp, 2006.