



Curso de Engenharia de Computação

ESTUDO SOBRE OTIMIZAÇÃO DE BANCO DE DADOS

Gustavo Buso Pontes

Campinas – São Paulo – Brasil

Dezembro de 2009



Curso de Engenharia de Computação

ESTUDO SOBRE OTIMIZAÇÃO DE BANCO DE DADOS

Gustavo Buso Pontes

Monografia apresentada à disciplina de Trabalho de Conclusão do Curso de Engenharia de Computação da Universidade São Francisco, sob a orientação do Prof. Ms. Marcio Henrique Zuchini, como exigência parcial para conclusão do curso de graduação.

Orientador: Prof. Mestre Marcio Henrique Zuchini.

Campinas – São Paulo – Brasil

Dezembro de 2009

ESTUDO SOBRE OTIMIZAÇÃO DE BANCO DE DADOS

Gustavo Buso Pontes

Monografia defendida e aprovada em 15 de Dezembro de 2009 pela **Banca Examinadora** assim constituída:

Prof. Ms. Marcio Henrique Zuchini

USF – Universidade São Francisco – Campinas – SP.

Prof. Carlos Eduardo Pagani

USF – Universidade São Francisco – Campinas – SP.

Prof. Ms. Cláudio Maximiliano Zaina

USF – Universidade São Francisco – Campinas – SP.

*A minha família, namorada e amigos, os quais me
ofereceram todo o apoio necessário.*

Agradecimentos

Agradeço primeiramente a Deus, que me deu a vida e a força para lutar.

Agradeço a meus pais e irmã por todo carinho e dedicação em todos os momentos da minha vida.

Agradeço a minha namorada por sempre demonstrar seu amor, apoio e dedicação.

Agradeço ao Prof. Marcio Henrique Zuchini, pelas críticas, ensinamentos e todo o apoio.

Sumário

Lista de Siglas	05
Lista de Figuras	06
Lista de Tabelas	07
Resumo	08
Abstract	09
1. INTRODUÇÃO	10
1.1 Contextualização	10
1.2 Objetivo	10
1.3 Estrutura do texto.....	11
2. REVISÃO BIBLIOGRÁFICA	12
2.1 Banco de Dados.....	12
2.2 Otimização (Tuning)	13
2.3 Ajuste de desempenho	14
2.4 Localização dos gargalos.....	14
2.5 Parâmetros ajustáveis	15
2.6 Ajuste de índices	16
3. PROJETO DESENVOLVIDO	18
3.1 Virtualização e VMware	18
3.2 PostgreSQL	19
3.3 Benchmark TPC-C	20
3.4 Benchmark SQL	20
3.4.1 Metodologia.....	21
3.4.2 Transações.....	22
3.4.3 Configuração do Benchmark SQL.....	23
4. IMPLEMENTAÇÃO	25

4.1 Máquinas virtuais	25
4.2 Simulação 1	26
4.2.1 Análise dos resultados	27
4.2.2 Otimização 1 do PostgreSQL	28
4.2.3 Análise dos resultados com a Otimização 1	30
4.3 Simulação 2	32
4.3.1 Análise dos resultados	32
4.3.2 Otimização 2 do PostgreSQL	33
4.3.3 Análise dos resultados com a Otimização 2	34
4.4 Análise das simulações	36
5. CONCLUSÃO	38
5.1 Contribuições	39
5.2 Trabalhos futuros	40
Referências Bibliográficas	41
Bibliografia Consultada	42
Apêndice 1 – Resultado detalhado da Simulação 1	43
Apêndice 2 – Resultado detalhado da Simulação 2	45

Lista de Siglas

ACID	Atomicidade, Consistência, Independência e Durabilidade.
CPU	<i>Central Processing Unit</i>
IDE	<i>Intelligent Drive Electronics</i>
JDBC	<i>Java Data Base Connectivity</i>
ODBC	<i>Open Data Base Connectivity</i>
OLTP	<i>On Line Transaction Processing</i>
RAM	<i>Random Access Memory</i>
SCSI	<i>Small Computer System Interface</i>
SGBD	Sistema Gerenciador de Banco de Dados
SQL	<i>Structured Query Language</i>
TPC-C	<i>Transaction Processing Performance Council</i>
TPC-H	<i>Transaction Processing Performance Ad-hoc</i>
TPMC	Tempo Médio de Processamento das Transações TPC-C
WAL	<i>Write Ahead Log</i>

Lista de Figuras

FIGURA 1: Diagrama da hierarquia das relações TPC-C no BenchmarkSQL.	21
FIGURA 2: Diagrama de entidade-relacionamento da base de dados de teste.....	22
FIGURA 3: Arquivo de configuração <i>PostgreSQL.properties</i>	24
FIGURA 4: Tela inicial do BenchmarkSQL.	25
FIGURA 5: Configuração dos recursos do servidor físico e das máquinas virtuais.....	26
FIGURA 6: Análise de desempenho dos Servidores 1 e 2 na Simulação 1.....	27
FIGURA 7: Comparação do desempenho da otimização do Servidor 3 em relação aos Servidores 1 e 2.....	30
FIGURA 8: Quantidade de transações realizadas por Servidor na Simulação 1	31
FIGURA 9: TPMC por Servidor na Simulação 1.....	31
FIGURA 10: Análise de desempenho dos Servidores 1 e 2 na Simulação 2.....	33
FIGURA 11: Comparação do desempenho da otimização do Servidor 3 em relação aos Servidores 1 e 2.....	34
FIGURA 12: Quantidade de transações realizadas por Servidor na Simulação 2..	35
FIGURA 13: TPMC por Servidor na Simulação 2.....	35
FIGURA 14: Comparação de desempenho entre os servidores da Simulação 1 e 2.	36
FIGURA 15: Comparação do desempenho da TPMC dos servidores na Simulação 1 e 2.....	37
FIGURA 16: Comparação da quantidade de transações realizadas nos servidores na Simulação 1 e 2.....	37

Lista de Tabelas

TABELA 1: Percentual de execução de cada transação no BenchmarkSQL.....	22
-------------------------------------------------------------------------	----

Resumo

A demanda por sistemas de banco de dados cresce continuamente e junto, o volume e a complexidade de dados que estes sistemas devem gerenciar. Este trabalho mostra a importância que a otimização de um sistema gerenciador de banco de dados pode exercer dentro de um ambiente que trabalha com processamento de transações em tempo real. Sendo realizada uma análise de desempenho da otimização do banco de dados PostgreSQL, com foco na configuração de seus parâmetros. Para a simulação e comparação dos testes de performance foi utilizada a ferramenta BenchmarkSQL que se baseia na metodologia TPC-C.

PALAVRAS-CHAVE: BANCO DE DADOS, OTIMIZAÇÃO, DESEMPENHO.

Abstract

The demand for database systems continually increases and with that, the volume and complexity of data these systems must manage. This work shows the importance of a database management system tuning which works on online processing transactions. An analysis of PostgreSQL database tuning performance has been made focusing on parameter configuration. For the simulation and comparison of performance tests, the tool BenchmarkSQL was used, based on TPC-C methodology.

PALAVRAS-CHAVE: DATABASE, TUNING, PERFORMANCE.

1 INTRODUÇÃO

1.1 Contextualização

Atualmente, encontram-se no mercado diversos Sistemas Gerenciadores de Banco de Dados (SGBD), os quais na maioria dos casos são instalados, configurados e utilizados com todos os seus parâmetros em valores padrão, desconsiderando o tipo de aplicação para o qual serão utilizados, o *hardware* e até mesmo o sistema operacional. Desta forma nem sempre se obtém o melhor desempenho do sistema, visto que diversos parâmetros podem ser considerados e ajustados.

Quando uma empresa não possui uma equipe técnica com um bom grau de treinamento e conhecimento do SGBD que utiliza, certamente não possuirá um sistema totalmente configurado e adaptado para as suas aplicações, o que acaba gerando uma baixa performance. Muitas vezes por falta desse conhecimento específico pode ocorrer um erro de planejamento, pois se investe em aquisição de novos equipamentos mais potentes, para suprir uma necessidade que poderia ser melhorada apenas com uma melhor administração dos recursos de *software* de seu SGBD.

1.2 Objetivo

Este trabalho tem por objetivo mostrar como um SGBD que irá trabalhar com aplicações OLTP (*On Line Transaction Processing*) pode se comportar ajustando-se parâmetros de configurações específicas e também de mudanças dos recursos de *hardware*. Intenciona-se mostrar que alguns ajustes podem trazer grandes ganhos de desempenho e que nem sempre o melhor caminho para melhoria de desempenho é o investimento em novos recursos de *hardware*.

1.3 Estrutura do texto

Este trabalho está dividido da seguinte maneira:

Capítulo 1 – Introdução contendo a contextualização e a estrutura do texto.

Capítulo 2 – Revisão Bibliográfica contendo os conceitos sobre banco de dados, otimização, o ajuste de desempenho, a localização dos gargalos, os parâmetros ajustáveis e o ajuste de índices.

Capítulo 3 – Projeto Desenvolvido contendo descrição da virtualização, do banco de dados e das ferramentas utilizadas.

Capítulo 4 – Implementação contendo a descrição das máquinas virtuais, a metodologia utilizada nas simulações, a técnica utilizada para otimização do sistema e a análise dos resultados.

Capítulo 5 – Conclusão contendo conclusão sobre os resultados obtidos, a contribuição e a proposta para os trabalhos futuros.

2 REVISÃO BIBLIOGRAFICA

2.1 Banco de dados

A necessidade por Sistemas de Gerenciamento de Bancos de Dados cresce continuamente. Junto com essa demanda, cresce também o volume de dados que estes sistemas devem gerenciar e a complexidade de suas aplicações. Os bancos de dados estão em toda parte e a maioria das pessoas interage diariamente, direta ou indiretamente com esses sistemas. Neste cenário, realizar operações com segurança, de forma eficiente, garantindo a integridade sobre estas grandes coleções de dados é uma questão fundamental, já que o desempenho de um SGBD é medido a partir de sua eficiência diante das consultas e alterações.

Segundo [SILBERSCHATZ 2006]: *“O principal objetivo de um banco de dados é fornecer um ambiente que seja tanto conveniente quanto eficiente para as pessoas usarem na recuperação e armazenamento das informações”*, oferecendo aos usuários apenas o resultado final através de uma visão abstrata dos dados e assim, ocultando os detalhes de como os dados são armazenados e mantidos.

[DATE 2003] define: *“Sistema Gerenciador de Banco de Dados é o software que manipula todos os acessos ao banco de dados”*. O SGBD é um software que funciona como uma interface entre o usuário e o banco de dados, ou seja, todas as solicitações dos usuários, como criação de tabelas, inserção de dados, recuperação de dados, são manipuladas pelo SGBD.

As principais funções desempenhadas por um SGBD incluem:

- **Processamento e Otimização de Consultas:** uma consulta expressa em linguagem de alto nível como SQL (*Structured Query Language*), deve primeiro, ser examinada, analisada, validada e otimizada, para só então ser executada;

- **Processamento de Transações:** a manipulação do banco de dados se dá por meio de transações. Uma transação é uma unidade lógica de processamento, formada por uma ou mais operações de acesso a dados. O SGBD deve garantir que as transações tenham um conjunto de propriedades conhecido como ACID (Atomicidade, Consistência, Independência e Durabilidade). Para tal, ele deve escalonar as transações, provendo controle de concorrência;
- **Recuperação de Falhas:** é responsabilidade de SGBD manter a consistência do banco de dados mesmo após falhas. Para fazer isso, o SGBD deve manter informações sobre as alterações executadas pelas transações em um arquivo de log. A partir do log é possível refazer ou desfazer os efeitos das transações;

2.2 Otimização (*Tuning*)

A avaliação de desempenho está presente em todos os momentos do ciclo de vida de um sistema computacional. Na hora de projetar, produzir ou implantar um sistema, o objetivo final é sempre o mesmo: escolher dentre diversas alternativas aquela que proporcione o melhor desempenho, com o menor custo possível.

Um banco de dados mal configurado ou que não tenha tido um bom projeto de desenvolvimento gera um grande desperdício de recursos tanto de *hardware* e *software* e essa improdutividade pode ser mais crítica em sistemas OLTP ou que necessitem de algum tipo de backup em tempo real, como por exemplo: sistemas de reserva de passagem aérea, sistemas bancários, sistemas de acompanhamento do mercado financeiro etc.

Após um banco de dados ter sido desenvolvido e estar em operação, o uso real das aplicações, das transações, das consultas e das visões revela fatores e áreas de problemas que podem não ter sido considerados durante o projeto de desenvolvimento.

As informações de entrada para o projeto físico podem ser revisadas por meio da coleta de estatísticas reais sobre os padrões de uso. A utilização dos recursos, bem como o

processamento interno do SGBD pode ser monitorado para revelar gargalos, tais como a disputa pelos mesmos dados ou dispositivos. Os volumes de atividades e os tamanhos dos dados podem ser melhor estimados. Portanto, é necessário monitorar e revisar o projeto físico do banco de dados constantemente, mostrando quais ajustes devem ser efetuados no sistema para melhor atender às necessidades dos usuários finais, propiciando a execução mais rápida do mesmo volume de transações sobre o mesmo *hardware* e *software*.

2.3 Ajuste de desempenho

No projeto de SGBD é comum descobrir que, quando uma aplicação é montada, ela fica mais lenta do que os desenvolvedores previam, ou trata menos transações por segundo. De acordo com [SILBERSCHATZ 2006]: “*Uma aplicação que leva um tempo excessivo para realizar as ações solicitadas pode causar, insatisfação do usuário e também se tornar inutilizável*”.

O ajuste de desempenho envolve ajustar vários parâmetros e opção de projeto a fim de melhorar seu desempenho para uma aplicação específica. Em um sistema de banco de dados, diversos aspectos podem ser ajustados, variando desde aspectos de alto nível, como os projetos de esquema de transações, até parâmetros do sistema como *buffer* de memória, limite de conexões e aspectos de *hardware*, como número de discos e tamanho de memória física.

Mas para que seja possível realizar o ajuste de um sistema, primeiro é preciso tentar descobrir quais são os gargalos do sistema e depois tentar eliminá-los da maneira mais eficaz possível.

2.4 Localização dos gargalos

O desempenho da maioria dos sistemas em geral é limitado principalmente pelo desempenho de alguns componentes, chamados de gargalos. Por exemplo, um programa pode gastar 80% do seu tempo em um pequeno *loop* dentro do código e 20% no restante do código, então o pequeno *loop* é considerado um gargalo. A melhoria do desempenho de um

componente que não seja um gargalo faz pouca diferença para melhorar a velocidade do sistema. No exemplo a melhoria de desempenho no restante do código não pode resultar em mais que 20% de melhoria do sistema, enquanto que a melhoria da velocidade do *loop* de gargalo, poderia resultar em um ganho significativo de desempenho.

Em um sistema balanceado nenhum componente isolado é um gargalo, quando um gargalo é removido pode acontecer que outro componente se torne um gargalo, pois o outro componente estava sendo subutilizado e talvez não estivesse preparado para uma nova demanda que surgiu com a solução do primeiro gargalo. [SILBERSCHATZ 2006] afirma: “*que utilizações de recursos em torno de 70% são consideradas boas, e as utilizações acima de 90% são consideradas excessivas e podem ser tratadas como um gargalo*”.

Pode-se concluir que em sistemas que contenham gargalos que não possam ser solucionados por algum motivo técnico ou financeiro, os componentes que não fazem parte do gargalo são subutilizados e talvez possam ser substituídos por componentes mais baratos com menor desempenho, gerando uma redução de custos.

2.5 Parâmetros ajustáveis

Os administradores de um banco de dados podem ajustar um SGBD em três níveis [SILBERSCHATZ, 2006].

O nível mais baixo é o de *hardware*. As opções para o ajuste de um sistema de banco de dados no nível de *hardware* podem incluir acrescentar discos rígidos de maior capacidade, aumentar a quantidade de memória, passar para um processador mais rápido ou aumentar a largura de banda. Para uma determinada função existem diversos tipos de *hardware* que são diferenciados de acordo com sua capacidade e performance, quanto melhor for a performance do dispositivo, maior será a tecnologia agregada e assim maior será o seu custo final. Por exemplo, um disco rígido IDE (*Intelligent Drive Electronics*) pode possuir a mesma capacidade de armazenamento de um disco SCSI (*Small Computer System Interface*), mas a performance e o custo do dispositivo SCSI será mais elevada.

O segundo nível consiste em parâmetros de configuração do sistema de banco de dados, como tamanho do *buffer* de controle de *checkpoints* (pontos de controle). O conjunto e tipo de parâmetros exatos que podem ser ajustados dependem do sistema específico de banco de dados, a maioria dos sistemas oferecem manuais com informações sobre quais parâmetros que podem ser ajustados. Sistemas de banco de dados bem projetados realizam o máximo de desempenho possível automaticamente, poupando os usuários e administradores desse trabalho.

O terceiro nível é o projeto de mais alto nível, englobando esquemas e transações. Pode-se ajustar o projeto do esquema, a criação de índices e a execução de transações de modo a melhorar o desempenho. O ajuste nesse nível é relativamente independente do tipo de SGBD.

Os três níveis de ajuste interagem entre si. Devemos considerar tal combinação quando ajustamos o sistema. Por exemplo, ajustar um nível mais alto pode resultar em uma alteração no gargalo de *hardware*, como do disco rígido para CPU (*Central Processing Unit*).

2.6 Ajuste de índices

Podemos ajustar os índices de um sistema de banco de dados para melhorar seu desempenho. Se for constatado que as consultas são um gargalo, pode-se acelerar seu processamento criando índices apropriados para relações. Se as atualizações são o gargalo, é provável que haja um excesso de índice que precisam ser alterados quando as relações são atualizadas. A remoção dos índices pode melhorar o desempenho das atualizações.

O uso de índices contribui para o processo de otimização do sistema. Os índices são estruturas opcionais mantidas pelo SGBD, que fornece a otimização e o acesso a seus dados. Para a criação de um índice é necessário especificar uma tabela e uma ou mais colunas a controlar. Uma vez criado um índice, o SGBD o mantém automaticamente quando os dados são criados, alterados e eliminados de suas tabelas. [DATE 2003], define as seguintes regras para ajuda na decisão de se criar um índice:

- Deve-se indexar colunas que tenham um intervalo de valores distintos. Se determinado valor da coluna de uma tabela é apresentado em 20% ou mais das tuplas, então a coluna é uma candidata para um índice;
- Se várias colunas são, continuamente referenciadas juntas nos predicados da declaração SQL, deve-se considerar a colocação delas em um índice;
- Ao construir um índice em uma tabela, se todos os valores de coluna índice são distintos, é aconselhável criá-lo como índice único, Encontrar modos de usar índices únicos ajuda o processo de otimização;

3 PROJETO DESENVOLVIDO

Este trabalho propõe uma análise da otimização do banco de dados PostgreSQL (seção 3.2), em diferentes ambientes de testes, através da ferramenta BenchmarkSQL (seção 3.4) que se baseia na metodologia TPC-C (*Transaction Processing Performance Council*) (seção 3.3).

Primeiramente foram criados dois ambientes de testes, sendo um com SGBD otimizado e outro não. Dentro de cada ambiente foram implementadas três máquinas virtuais com o *software* VMware (seção 3.1), cada uma com uma configuração específica de *hardware*, mas com a mesma configuração de sistema operacional.

Após a criação das máquinas virtuais, foi realizado a instalação e configuração do banco de dados PostgreSQL, que foi submetido a uma carga de testes de desempenho com o *software* BenchmarkSQL nos diferentes ambientes com as diferentes configurações de otimização. O *software* simulou uma carga de dados e a execução de determinado conjunto de transações, por um determinado período de tempo. Nos dois ambientes virtuais o SGBD trabalhou com transações OLTP em duas situações: com uma quantidade razoável de transações, aproximadamente 12.000 e outro com uma grande quantidade de transações superior a 200.000.

Depois de executado todas as cargas de testes em cada máquina virtual, coletou-se os dados obtidos e realizou uma análise de comparação do desempenho do PostgreSQL. Observando seu comportamento em cada máquina virtual e em especial no ambiente onde foi realizada a otimização do SGDB.

3.1 Virtualização e VMware

Virtualização é o processo de executar vários sistemas operacionais em um único equipamento. Uma máquina virtual é um ambiente operacional completo que se comporta

como se fosse um computador independente. Com a virtualização, um servidor pode manter vários sistemas operacionais em uso [VMware 2008].

Neste trabalho foi utilizado o software VMware Workstation 6.5, que permite trabalhar com um ou mais sistemas operacionais simultaneamente num ambiente isolado, criando máquinas virtuais completas dentro de um computador físico que está utilizando um sistema operacional totalmente distinto [VMware 2008].

3.2 PostgreSQL

O PostgreSQL é um Sistema Gerenciador de Banco de Dados Objeto Relacional, desenvolvido pelo Departamento de Ciência da Computação da Universidade da Califórnia em Berkeley. Suporta grande parte do padrão SQL:2003, além de serem oferecidas muitas funcionalidades modernas que existem nos principais SGBD comerciais, como:

- comandos complexos;
- chaves estrangeiras;
- gatilhos;
- visões;
- integridade transacional;
- controle de simultaneidade multiversão;

Devido à sua licença aberta, o PostgreSQL pode ser utilizado, modificado e distribuído por qualquer pessoa para qualquer finalidade, seja privada, comercial ou acadêmica, livre de encargos. Para este trabalho foi utilizado a versão 8.4.1.

3.3 Benchmark TPC-C

O *benchmark* TPC-C é considerado um dos principais metodologia da indústria para avaliação da performance de sistemas OLTP. Encarrega-se de testar a maior parte das funcionalidades de um banco de dados, como consultas e atualizações [TPC-C 2009].

TPC-C mede a taxa de transferência de transações de negócios mesclando transações de apenas leitura com transações que fazem intensivas atualizações. O teste quantifica quantas novas transações de pedidos um sistema pode absorver por minuto enquanto esse mesmo sistema executa, simultaneamente, outros quatro tipos de transações.

3.4 BenchmarkSQL

O BenchmarkSQL é uma ferramenta de código aberto baseada na metodologia TPC-C é desenvolvida em linguagem de programação Java. Utiliza drivers JDBC (*Java Database Connectivity*) para se comunicar com o SGBD e isto o torna compatível com a grande maioria dos sistemas de bancos de dados.

Por ser uma ferramenta baseada na metodologia TPC-C, que o objetivo é a avaliação do desempenho de determinado sistema de banco de dados, através da execução de diversas transações do tipo OLTP, mistura transações de apenas leitura com transações que fazem intensivas atualizações no banco de dados.

3.4.1 Metodologia

O BenchmarkSQL é composto de uma série de transações projetadas para simular um típico sistema de aquisição de produtos, com funcionalidades genéricas que se enquadram para qualquer tipo de negócio que trabalhem com transações *online*.

O ambiente simula um fornecedor atacadista que possui determinado número de territórios de vendas e lojas associadas. Conforme a empresa cresce, novas lojas e territórios são criados. Cada território tem capacidade para atender 3000 clientes. Existem também os armazéns regionais, sendo que cada armazém atende 10 territórios. Todas as lojas mantêm um estoque de 100.000 produtos que são vendidos pela empresa. Os detalhes do ambiente de simulação podem ser visto nas Figuras 1 e 2:

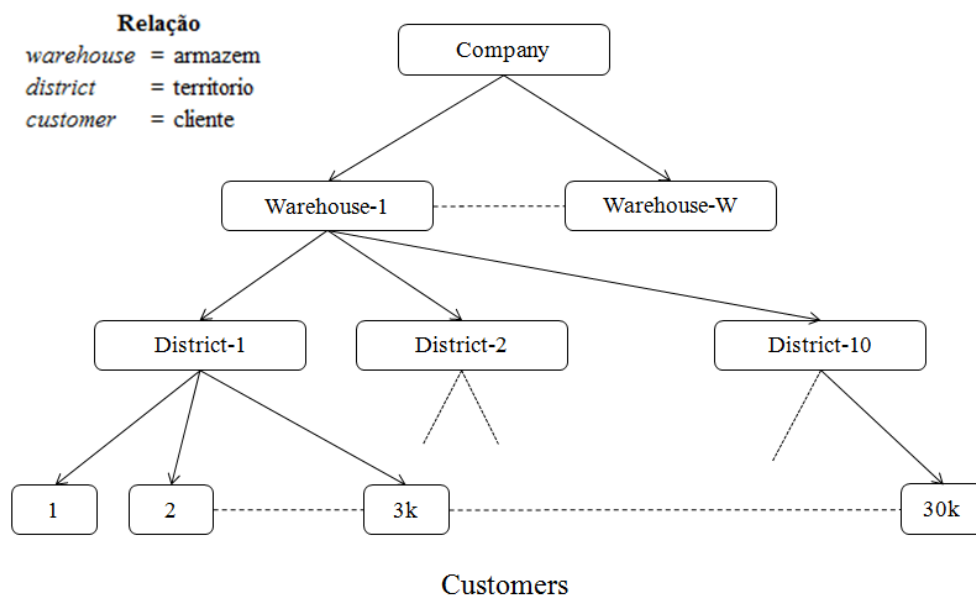


FIGURA 1: Diagrama da hierarquia das relações TPC-C no BenchmarkSQL.

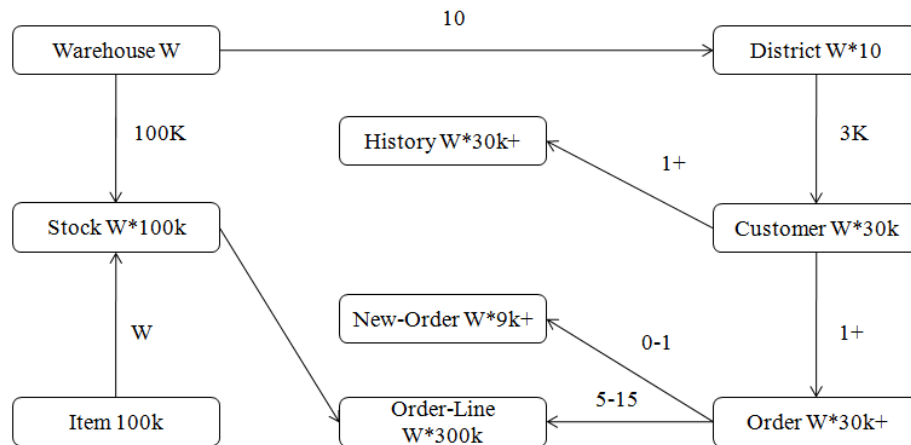


FIGURA 2: Diagrama de entidade-relacionamento da base de dados de teste.

3.4.2 Transações

O BenchmarkSQL define cinco tipos de transações que devem ser submetidas contra o SGBD durante a carga de teste: *New-Order*, *Payment*, *Order-Status*, *Delivery* e *Stock-Level*.

A tabela 1 mostra o percentual de execução de cada transação definido de acordo com as especificações do TPC-C, sendo possível sua alteração dentro do BenchmarkSQL, assim sendo possível obter uma simulação mais próxima da aplicação real e focando nas transações mais específicas para cada tipo de simulação [TPC-C 2009].

TABELA 1: Percentual de execução de cada transação no BenchmarkSQL.

<i>New-Order</i>	<i>Payment</i>	<i>Order-Status</i>	<i>Delivery</i>	<i>Stock-Level</i>
45%	43%	4%	4%	4%

- ***New-Order***: A transação *New-Order* representa a entrada de uma nova ordem no sistema. Considerada uma transação leve, não consumindo muitos recursos computacionais e com alto índice de execução;

- **Payment:** A transação *Payment* atualiza o crédito de um cliente e reflete este pagamento nas estatísticas da tabela de territórios (*DISTRICT*). Considerada uma transação leve e com alto índice de execução;
- **Order-Status:** A transação *Order-Status* verifica o estado da última ordem de pedido solicitada por um cliente. Transação de apenas leitura e considerada mediana em relação ao consumo de recursos;
- **Delivery:** A transação *Delivery* envolve o processamento em lote de dez novas ordens. Cada ordem é processada por completo, envolvendo operações de leitura e escrita na tabelas do banco de dados. Considerada uma transação leve, pois possui baixa frequência de execução;
- **Stock-Level:** A transação *Stock-Level* fornece o número de itens vendidos recentemente e que possuem quantidade em estoque abaixo do limite especificado. Esta transação representa uma operação pesada, ou seja, que demanda grande quantidade de recursos computacionais.

3.4.3 Configuração do BenchmarkSQL

Consiste em cinco etapas: Configuração do Acesso ao SGBD, Criação das Tabelas no SGBD, Carga das Tabelas, Criação de Índices e Execução do BenchmarkSQL [BenchmarkSQL 2006].

- **Configuração do Acesso ao SGBD:** Antes de realizar a execução de qualquer atividade no BenchmarkSQL, primeiramente foi necessário fazer a configuração do acesso ao SGBD utilizado. Essa configuração foi realizada dentro do arquivo PostgreSQL.properties, que contém o driver ODBC (Open Data Base Connectivity) do SGBD, caminho de acesso, usuário e senha. Como mostra a Figura 3:

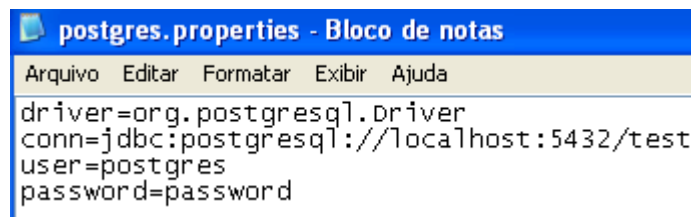


FIGURA 3: Arquivo de configuração *PostgreSQL.properties*

- **Criação das Tabelas:** Nesta etapa foram criadas as tabelas dentro do SGBD, executando um arquivo que faz parte do software BenchmarkSQL, chamado `runSQL.bat` para o sistema operacional Windows ou `runSQL.sh` para o sistema Unix, mais o comando `sqlTableCreates`. Comando:

runSQL PostgreSQL.properties sqlTableCreates

- **Carga das Tabelas:** Para a carga das tabelas no SGBD foi executado o arquivo `loadData.bat` para o sistema operacional Windows ou `loadData.sh` para o sistema Unix. Que gerou uma base de dados aleatórios nas tabelas do método TPC-C, aproximadamente 600.000 registros foram inseridos. Comando:

loadData PostgreSQL.properties

Por padrão o BenchmarkSQL esta preparado para trabalhar apenas com uma *Warehouse*, sendo possível aumentar o número de *Warehouses* do ambiente, através do parâmetro `numWarehoues=N`, no final do comando `loadData`. Comando:

loadData PostgreSQL.properties numWarehouses=10

- **Criação de Índices:** Nesta etapa foram criados os índices e as chaves primárias que identificam as tabelas, sendo apenas necessário executar o comando `sqlIndexCreates`. Comando:

runSQL PostgreSQL.properties sqlIndexCreates

- **Execução do BenchmarkSQL:** Após conclusão das etapas de configurações, foi executado o programa BenchmarkSQL através do comando:

runBenchmark PostgreSQL.properties

A Figura 4, mostra a tela inicial do BenchmarkSQL, com as opções de configuração do banco de dados e o nome de usuário e senha.

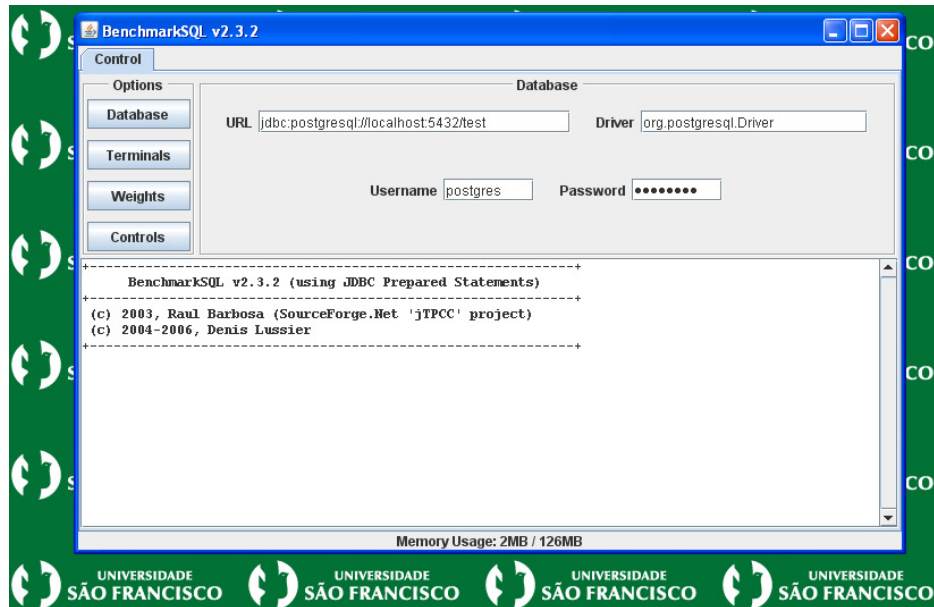


FIGURA 4: Tela inicial do BenchmarkSQL.

4 IMPLEMENTAÇÃO


Este capítulo tem por objetivo apresentar os detalhes da implementação dos dois ambientes de simulação que foram utilizados e uma análise dos resultados de desempenho do SGBD dentro da cada ambiente.

4.1 Máquinas virtuais

Para a realização deste trabalho utilizou-se um servidor físico como hospedeiro, em que foi instalado *software* VMware, dentro do *software* foram criadas três máquinas virtuais, chamadas de Servidor 1, Servidor 2 e Servidor 3. Os Servidores 1 e 2 possuem as mesmas configurações de *software*: sistema operacional Windows XP e banco de dados PostgreSQL com sua configuração padrão, sendo diferenciados pela configuração de *hardware* que torna o Servidor 2 superior ao Servidor 1, devido o fato de possuir uma maior quantidade de memória RAM (*Random Access Memory*)

No caso do Servidor 3 que possui a mesma configuração de *hardware* do Servidor 1, foram realizadas alterações nas configurações padrões do banco de dados PostgreSQL, criando uma otimização voltada para operar com transações OLTP. Essas configurações estão descritas em detalhes na seção 4.2.2. A Figura 5 mostra as configuração dos recursos do servidor físico e das máquinas virtuais:

Hospedeiro - Windows® Vista					
	Processador	Memória	Disco	Sistema de Arquivos	
Servidor Físico	Pentium Dual Core 1,46Ghz	3.0 Gb	120.0 Gb	NTFS	



Máquina Virtual - Windows® XP + PostgreSQL®					
	Processador	Memória	Disco	Sistema de Arquivos	PostgreSQL Otimizado
Servidor 1	Pentium Dual Core 1,46Ghz	512 Mb	8.0 Gb	NTFS	NÃO
Servidor 2	Pentium Dual Core 1,46Ghz	1024 Mb	8.0 Gb	NTFS	NÃO
Servidor 3	Pentium Dual Core 1,46Ghz	512 Mb	8.0 Gb	NTFS	SIM

FIGURA 5: Configuração dos recursos do servidor físico e das máquinas virtuais.

4.2 Simulação 1

Após a etapa de criação dos três servidores virtuais, foi criada a primeira simulação de desempenho, que consistiu em criar um ambiente que simulou um número razoável de transações OLTP nos três servidores virtuais por um período de 10 minutos, através do BenchmarkSQL. Para esta etapa foi utilizado a relação de 1 *warehouse* para 10 terminais.

O primeiro passo foi realizar a carga da base de dados no PostgreSQL, essa carga consiste na criação de um banco de dados chamado TEST, que contém nove tabelas com aproximadamente 600.000 tuplas e dentro dessas nove tabelas foram geradas várias transações do tipo: *New-Order*, *Payment*, *Order-Status*, *Delivery* e *Stock-Leve*. Importante ressaltar que para garantir um melhor resultado das avaliações, sempre que uma simulação é

iniciada a base de dados antiga é excluída e um novo banco é criado novamente com mesmas características.

4.2.1 Análise dos resultados

Para poder mensurar o desempenho de um SGBD pelo BenchmarkSQL, foram utilizadas duas métricas de avaliação de performance: a primeira foi a Quantidade de Transações que foram processadas em um período de tempo e a segunda o Tempo Médio de Processamento das Transações TPC-C (TPMC) que será expresso em milissegundos (ms).

Primeiramente realizou-se uma análise do desempenho dos Servidores 1 e 2, conforme segue na Figura 6 :

	Quantidade Transações	Média TPMC (ms)	Duração (min)
Servidor 1	12.152	496,59	10
Servidor 2	9.107	657,85	10

	Quantidade Transações	Média TPMC (ms)
Servidor 1 X Servidor 2	-25,1%	32,5%

FIGURA 6: Análise de desempenho dos Servidores 1 e 2 na Simulação 1.

Observa-se que no Servidor 2 foram realizadas 25,1% menos transações que o Servidor 1 e que a TPMC do Servidor 2, foi 32,5% maior que no Servidor 1.

Os resultados das simulações detalhadas por cada servidor e seus terminais, encontra-se no Apêndice 1.

4.2.2 Otimização 1 do PostgreSQL

Para a realização da simulação no Servidor 3, foi desenvolvida uma configuração otimizada para o banco de dados PostgreSQL, com o objetivo de melhorar o desempenho do SGBD em um ambiente que necessite trabalhar com muitas transações OLTP, mas com comandos não muito complexos. Para isso foram alterados alguns parâmetros considerados mais importantes para um sistema que será submetido a este ambiente.

Todas as alterações foram feitas diretamente no arquivo de configuração do PostgreSQL, *postgresql.conf*. A baixo seguem as alterações que foram realizadas nos parâmetros e o motivo dessas alterações. As informações complementares a respeito de cada parâmetro estão disponíveis em maiores detalhes no manual do SGBD [PostgreSQL 2009].

- ***max_connections***: número máximo de conexões concorrentes que podem fazer acesso ao banco de dados, pode ser considerado um dos parâmetros mais importantes para se obter um bom desempenho em transações OLTP. Seu valor padrão é 100, mas foi alterado para 130.
- ***shared_buffers***: número de *buffers* da memória compartilhada que é utilizada pelo SGBD. Este parâmetro é considerado um dos mais importantes dentro do PostgreSQL. Seu valor padrão é 32Mb, como foi utilizado um servidor dedicado apenas para SGBD com 512Mb de memória disponível, com um sistema operacional que não exige um grande consumo de memória, seu valor foi alterado para 128Mb, 25% da memória disponível.
- ***wal_buffers***: este parâmetro é responsável pelo tamanho do número de *buffers* que podem ser utilizados pelo WAL (*Write Ahead Log*) do sistema, garantindo que os registros possam ser recuperados no caso de falha em alguma transação. Como o ambiente que está sendo simulado realiza muitas transações de escrita e leitura e gera uma grande quantidade *Logs*, esse parâmetro teve seu valor padrão de 64Kb alterado para 128Kb. Entretanto, é importante ressaltar que

aumentar muito o valor deste parâmetro pode resultar em perda de dados durante uma possível queda de energia, por exemplo.

- ***maintenance_work_mem***: quantidade de memória disponível para operações de manutenção, como por exemplo, o comando *Vacuum*. O comando *Vacuum* recupera a área de armazenamento ocupada pelas tuplas excluídas. Na operação normal do PostgreSQL as tuplas excluídas, ou tornadas obsoletas por causa de uma atualização, não são fisicamente removidas da tabela, permanecem presentes até o comando *Vacuum* ser executado. Como o ambiente simulado faz um número grande de alterações nas tuplas, foi realizada uma alteração no parâmetro *maintenance_work_mem* que tem seu valor padrão definido em 16Mb e foi alterado para 48Mb, com o objetivo de aumentar a velocidade de processamento do comando *Vacuum*.
- ***effective_cache_size***: este parâmetro define o quanto de memória RAM será alocada para o sistema, sendo a configuração que agrega um maior desempenho ao SGBD, pois evita a constante leitura das tabelas e índices, dos arquivos do disco rígido. Por exemplo, se uma tabela do banco de dados possui 20Mb, e o tamanho para esta configuração limita-se a 8Mb, o SGBD irá carregar a tabela por etapas, em partes, até a conclusão total da operação. Por isso este parâmetro impacta diretamente o aumento da performance do banco de dados, pois diminui consideravelmente as operações entrada e saída de disco. O valor padrão para este parâmetro é 128Mb, mas como esta sendo utilizado um servidor dedicado apenas ao SGBD, foi alterado para 50% da memória RAM, ou seja, 256Mb.

4.2.3 Análise dos resultados com a Otimização 1

Após a realização da otimização do banco de dados PostgreSQL, foi realizado a simulação dentro do Servidor 3 e uma nova análise dos resultados comparando a performance do Servidor 3 contra o Servidor 1 e 2. Conforme mostrado na Figura 7:

	Quantidade Transações	Média TPMC (ms)	Duração (min)
Servidor 1	12.152	496,59	10
Servidor 2	9.107	657,85	10
Servidor 3	14.082	422,92	10

	Quantidade Transações	Média TPMC (ms)
Servidor 1 X Servidor 3	15,9%	-14,8%
Servidor 2 X Servidor 3	54,6%	-35,7%

FIGURA 7: Comparação do desempenho da otimização do Servidor 3 em relação aos Servidores 1 e 2.

Percebe-se que o Servidor 3 teve um desempenho superior em 15,9% no número de transações que o Servidor 1 e a TPMC menor em 14,8%, sendo que os dois servidores possuem a mesma configuração de *hardware*.

A análise entre o Servidor 2 e o Servidor 3, mostra que o Servidor 3 teve um desempenho no número de transações superior a 54,6% e a média TPMC menor em 35,7%, sendo que o Servidor 2 esta com uma configuração de *hardware* superior.

Os gráficos das Figuras 8 e 9, mostram que a relação entre a quantidade de transações executadas e a métrica TPMC são inversamente proporcionais, quanto menor for a TPMC do SGBD, maior será o número de transações realizadas pelo sistema.

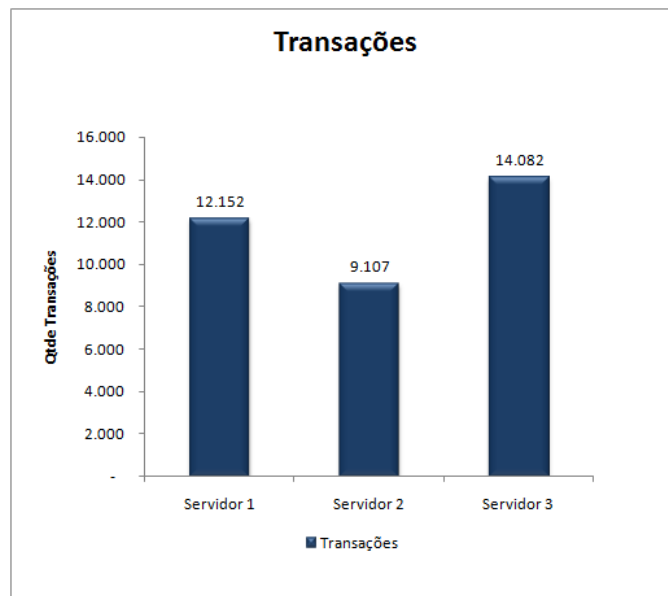


FIGURA 8: Quantidade de transações realizadas por Servidor na Simulação 1.

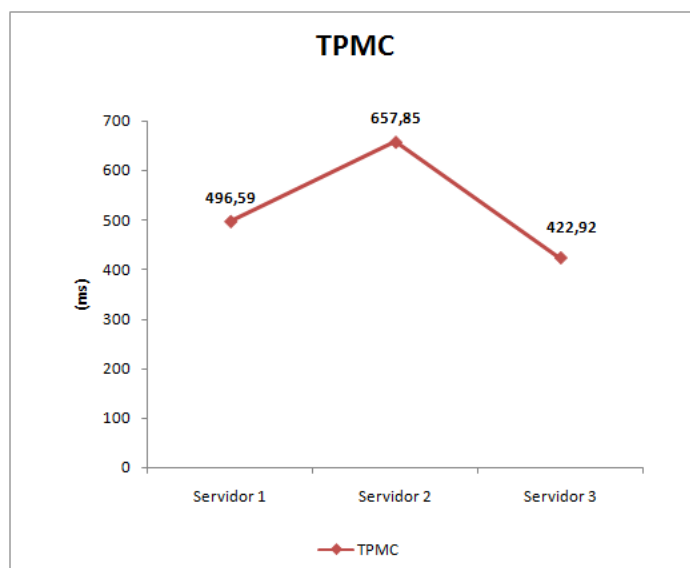


FIGURA 9: TPMC por Servidor na Simulação 1.

4.3 Simulação 2

Nesta etapa criou-se um ambiente que simulou um grande número de transações OLTP nos três servidores virtuais, por um período de 20 minutos, através do BenchmarkSQL. Para isto foi utilizada uma relação de 10 *warehouses* para 50 terminais.

O primeiro passo foi realizar a carga da base de dados no PostgreSQL, essa carga consiste na criação de um banco de dados chamado TEST2 que contém nove tabelas com aproximadamente 3.000.000 tuplas e dentro dessas nove tabelas foram geradas várias transações do tipo: *New-Order*, *Payment*, *Order-Status*, *Delivery* e *Stock-Leve*. Importante ressaltar que para garantir um melhor resultado das avaliações, sempre que uma simulação é iniciada o banco de dados antigo é excluído e um novo banco é criado novamente com mesmas características.

4.3.1 Análise dos resultados

Para esta simulação foram utilizadas a mesmas métricas da Simulação 1: quantidade de transações e a TPMC.

Observa-se que na Figura 10, que o número de transações realizadas pelos Servidores 1 e 2 foi aproximadamente 2.000 vezes maior que número de transações realizadas na Simulação 1 .

	Quantidade Transações	Média TPMC (ms)	Duração (min)
Servidor 1	253.222	76,39	20
Servidor 2	205.916	98,56	20

	Quantidade Transações	Média TPMC (ms)
Servidor 1 X Servidor 2	-18,7%	29,0%

FIGURA 10: Análise de desempenho dos Servidores 1 e 2 na Simulação 2.

A Figura 10 mostra que no Servidor 2 foram realizadas 18,7% a menos de transações e que a TPMC no Servidor 2 foi 29,0% maior que no Servidor 1.

Os resultados das simulações detalhadas por cada servidor, *warehouses* e seus terminais, encontra-se na seção Apêndice 2.

4.3.2 Otimização 2 do PostgreSQL

Para a realização da Simulação 2 no Servidor 3, foi mantida a mesma configuração de otimização da Simulação 1, pois todos os parâmetros já estavam ajustados na configuração máxima dentro do SGBD, com exceção do parâmetro *max_connections* que será detalhado a seguir.

Como o parâmetro *max_connections* é considerado um dos mais importantes para transações OLTP, e a Simulação 2 executou com um número muito superior de transações que a Simulação 1, este parâmetro teve seu valor aumentado em duas vezes, passando do valor padrão de 100 conexões para 200.

4.3.3 Análise dos resultados com a Otimização 2

Após a realização da otimização do sistema do PostgreSQL, foi realizado a simulação dentro do Servidor 3 e uma nova análise dos resultados comparando a performance do Servidor 3 contra o Servidor 1 e 2. Conforme mostrado na Figura 11:

	Quantidade Transações	Média TPMC (ms)	Duração (min)
Servidor 1	253.222	76,39	20
Servidor 2	205.916	98,56	20
Servidor 3	272.547	52,31	20

	Quantidade Transações	Média TPMC (ms)
Servidor 1 X Servidor 3	7,6%	-31,5%
Servidor 2 X Servidor 3	32,4%	-46,9%

FIGURA 11: Comparação do desempenho da otimização do Servidor 3 em relação aos Servidores 1 e 2.

Percebe-se que o Servidor 3 teve um desempenho superior a 7,6% no número de transações que o Servidor 1 e a TPMC menor em 31,5%, sendo que os dois servidores possuem a mesma configuração de *hardware*.

A análise entre o Servidor 2 e o Servidor 3, mostra que o Servidor 3 teve um desempenho no número de transações superior a 32,4% no Servidor 2 e a TPMC menor em 46,9,7%, sendo que o Servidor 2 esta com uma configuração de *hardware* superior.

Os gráficos das Figuras 12 e 13, mostram que a relação entre a quantidade de transações executadas e a TPMC também continuam inversamente proporcionais, seguindo a mesma tendência da Simulação 1.

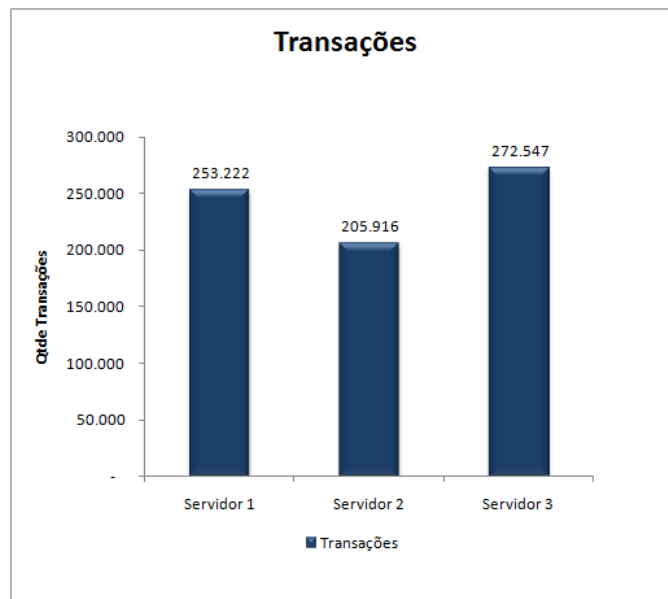


FIGURA 12: Quantidade de transações realizadas por Servidor na Simulação 2.

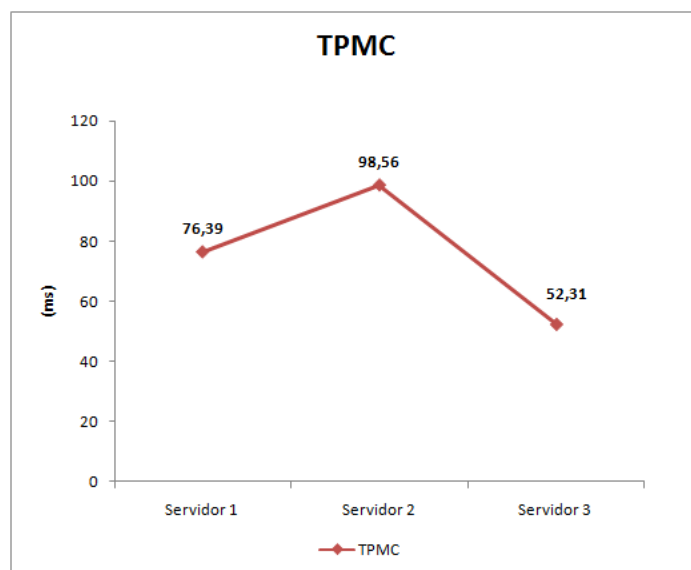


FIGURA 13: TPMC por Servidor na Simulação 2.

4.4 Análise das simulações

Percebe-se que existe uma perda de desempenho do Servidor 1 e Servidor 2 em comparação ao Servidor 3, nas duas simulações. Percebe-se também que essa perda é menor na Simulação 2 que trabalha com um número muito maior de transações e um período maior de tempo. Enquanto na Simulação 1 o Servidor 3 realiza 15,9% a mais de transações em comparação com o Servidor 1, na Simulação 2 esta diferença diminui para 7,6%, essa diferença no caso do Servidor 2 vai de 54,6% na Simulação 1, para 32,4% na Simulação 2. Na Figura 14, é possível visualizar essas diferenças.

	Simulação 1			Simulação 2		
	Quantidade Transações	Média TPMC (ms)	Duração (min)	Quantidade Transações	Média TPMC (ms)	Duração (min)
Servidor 1	12.152	496,59	10	253.222	76,39	20
Servidor 2	9.107	657,85	10	205.916	98,56	20
Servidor 3	14.082	422,92	10	272.547	52,31	20

	Quantidade Transações	Quantidade Transações
Servidor 1 X Servidor 3	15,9%	-14,8%
Servidor 2 X Servidor 3	54,6%	-35,7%

	Quantidade Transações	Média TPMC (ms)
Servidor 1 X Servidor 3	7,6%	-31,5%
Servidor 2 X Servidor 3	32,4%	-46,9%

FIGURA 14: Comparação de desempenho entre os servidores da Simulação 1 e 2 .

A TPMC do Servidor 3 obteve seu melhor desempenho de 52,31ms na Simulação 2, que exige uma maior capacidade de processamento de transações em um período mais longo de tempo. Mostrando que foi realizada uma configuração de otimização adequada para o ambiente de testes em que o SGBD foi submetido. Conforme mostra o gráfico da Figura 15.

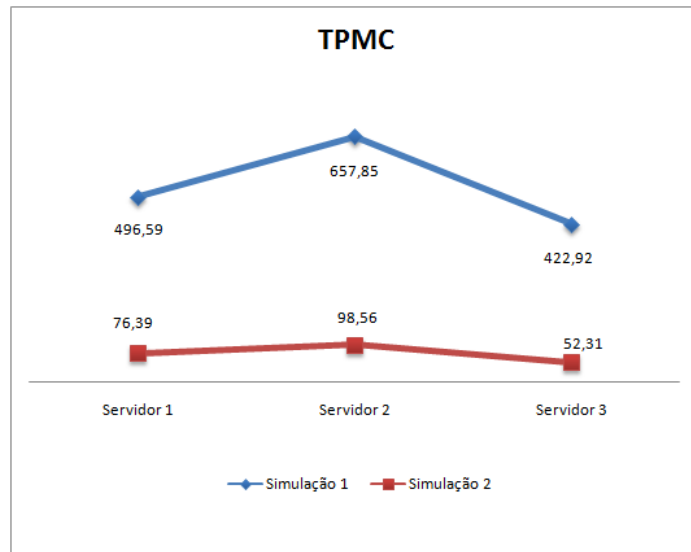


FIGURA 15: Comparação do desempenho da TPMC dos servidores na Simulação 1 e 2.

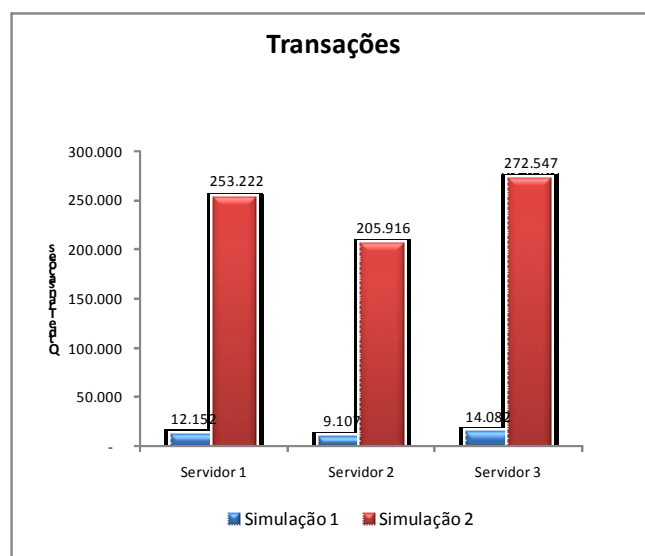


FIGURA 16: Comparação da quantidade de transações realizadas nos servidores na Simulação 1 e 2.

5 CONCLUSÃO

Este trabalho teve como objetivo mostrar a importância da otimização de um SGBD em um ambiente OLTP, ou seja, que trabalha com uma grande quantidade de transações concorrentes e consultas simples. Através da ferramenta BenchmarkSQL que simula um ambiente de teste através da metodologia TPC-C, sendo possível mostrar as diferenças de comportamento do SGBD nos dois ambientes de testes propostos.

Percebe-se que para um ambiente OLTP existe um gargalo nas configurações padrões do PostgreSQL e que para a otimização dos parâmetros de configuração foi levado em consideração alguns fatores específicos que envolveram: o tamanho de memória disponível no servidor, quantidade de conexões que podem ser realizadas no banco de dados simultaneamente, tamanho da memória disponível para operações de manutenção e configuração dos parâmetros de *log*. Concluindo-se que as otimizações foram realizadas de forma balanceada, pois o gargalo não estava em um parâmetro isolado.

Com relação ao SGBD PostgreSQL pode-se afirmar que sua configuração padrão comporta-se de forma eficiente em grandes bases de dados e quando executa funções complexas, só que tal configuração não exerce o mesmo desempenho em um ambiente OLTP, assim, sendo de extrema importância sua otimização para este tipo de ambiente. Também foi comprovado que o servidor que estava com banco de dados PostgreSQL otimizado, teve uma performance muito superior aos outros servidores, tanto no ambiente de Simulação 1 que trabalhou com uma quantidade menor transações e tanto para a Simulação 2 que trabalhou com uma quantidade maior de transações em um período maior de tempo.

Conclui-se que a realização de uma otimização correta para a o tipo de aplicação que o SGBD irá trabalhar pode resultar em ganho significativo de desempenho, sendo de extrema importância possuir um bom conhecimento técnico do SGBD que será utilizado. Também se conclui que liberar demasiadamente recursos computacionais do sistema para o SGBD nem sempre resulta em aumento de desempenho, muito pelo contrário, às vezes provoca perda de desempenho. Podendo afirmar que a otimização de um SGBD em uma corporação pode

resultar em uma redução substancial de custos, pois pode-se ter uma melhoria de performance do sistema e um ganho de produtividade dos usuários finais, sem a necessidade de investimento em novos recursos de *hardware* e *software*.

Otimização de banco de dados ou simplesmente *tuning* pode ser definido com um processo continuado de obtenção de dados, que analisados, permitirão decidir quais ajustes devem ser efetuados no sistema para melhor atender às necessidades dos usuários, propiciando a execução mais rápida do mesmo volume de transações sobre o mesmo *hardware* e *software*.

5.1 Contribuições

Através deste trabalho foi possível comprovar que um SGBD mal configurado gera um grande desperdício de recursos, mostrando a importância e necessidade dos benefícios da otimização no PostgreSQL dentro do ambiente OLTP, gerando um ganho de performance significativo sobre os mesmos recursos de *hardware*.

Assim, após a uma análise e conclusão, podemos citar abaixo as contribuições que este trabalho oferece para o leitor que deseja conhecer mais sobre a otimização de banco de dados:

- Que tipos de parâmetros podem ser ajustados no PostgreSQL, quando se trabalha com transações OLTP e porque desses ajustes;
- Implementação de uma forma de análise de performance para sistemas que trabalhem com transações OLTP.

5.2 Trabalhos futuros

Para este trabalho foi utilizada a ferramenta BenchmarkSQL que utiliza o TPC-C que é uma das metodologias de análise de performance para SGBD da instituição TPC. A proposta seria o desenvolvimento de um trabalho sobre os outros *benchmarks* da TPC, como por exemplo o TPC-H (*Transaction Processing Performance Ad-hoc*) que simula um sistema de suporte a decisão para grandes volumes de dados em tempo real.

Outra sugestão seria o desenvolvimento deste mesmo trabalho com o BenchmarkSQL em outros SGBD, que estão disponíveis no mercado e se possível a realização de uma comparação entre o desempenho de cada fabricante.

Referências Bibliográficas

[SILBERSCHATZ 2006] SILBERSCHATZ, Abraham; SILBERSCHATZ, Henry; SUDARSHAN, S. Sistema de Banco de Dados. 5 ed. São Paulo: Pearson Makron Books, 2006.

[DATE 2003] DATE, C. J. Introdução a Sistemas de Banco de Dados. 5 ed. Rio de Janeiro: Campus, 2003.

[PostgreSQL 2009] Documentação do PostgreSQL 8.0.0. Disponível online em <http://www.postgresql.org.br/docs>

[BenchmarkSQL 2006] Documentação do BenchmarkSQL User Guide. Disponível online em <http://sourceforge.net/projects/benchmarksql/>

[TPC-C 2009] TPC BENCHMARK™ C. Standard Specification Revision 5.10. Disponível em http://www.tpc.org/tpcc/spec/tpcc_current.pdf

[VMware 2008] VMware Workstation User's Manual. Disponível online em http://www.vmware.com/pdf/ws7_manual.pdf

Bibliográfica Consultada

Guest Operating System Installation Guide. Disponível online em
http://www.vmware.com/pdf/GuestOS_guide.pdf

RODRIGUES, Marcelo Nunes. Análise de Desempenho de Sistemas OLTP – Utilizando Benchmark TPC-C. Centro de Informática – Universidade Federal de Pernambuco: Pernambuco 2006.

LEMKE, Vilson Zarnott. Análise Comparativa dos Parâmetros Envolvidos no Ajuste de Desempe dos Bancos de Dados Oracle 9i e Interbase 6.5. Universidade Luterana do Brasil: Gravataí, 2002.

Apêndice 1 - Resultado detalhado da Simulação 1

As tabelas abaixo, mostram o resultado das simulações detalhadas por cada servidor e seus terminais, para a Simulação 1:

SERVIDOR 1

Terminal	Quantidade Transações	Média TPMc (ms)	Duração (min)
Term-01	1.198	503,42	10
Term-02	1.246	484,04	10
Term-03	1.236	488,37	10
Term-04	1.148	525,60	10
Term-05	1.173	514,28	10
Term-06	1.237	487,70	10
Term-07	1.232	490,04	10
Term-08	1.196	504,91	10
Term-09	1.248	483,76	10
Term-10	1.238	487,45	10
Total	12.152	496,59	10

SERVIDOR 2

Terminal	Quantidade Transações	Média TPMc (ms)	Duração (min)
Term-01	991	604,09	10
Term-02	830	721,79	10
Term-03	988	606,48	10
Term-04	885	677,49	10
Term-05	874	685,46	10
Term-06	843	710,59	10
Term-07	950	630,77	10
Term-08	875	684,20	10
Term-09	969	618,52	10
Term-10	902	664,28	10
Total	9.107	657,85	10

SERVIDOR 3

Terminal	Quantidade Transações	Média TPMc (ms)	Duração (min)
Term-01	1.401	425,28	10
Term-02	1.415	420,73	10
Term-03	1.483	401,70	10
Term-04	1.406	424,17	10
Term-05	1.391	427,51	10
Term-06	1.391	427,68	10
Term-07	1.445	412,29	10
Term-08	1.362	437,54	10
Term-09	1.394	426,83	10
Term-10	1.394	427,60	10
Total	14.082	422,92	10

Apêndice 2 - Resultado detalhado da Simulação 2

As tabelas abaixo, mostram o resultado das simulações detalhadas por cada servidor e seus terminais, para a Simulação 2:

SERVIDOR 1

Terminal	Quantidade Transações	Média TPMc (ms)	Duração (min)
Term-01	5.239	73,92	20
Term-02	8.242	62,81	20
Term-03	5.989	79,38	20
Term-04	3.863	76,29	20
Term-05	1.111	132,35	20
Term-06	377	120,60	20
Term-07	10.900	52,88	20
Term-08	347	135,17	20
Term-09	103	290,50	20
Term-10	3.798	74,28	20
Term-11	4.766	69,27	20
Term-12	10.938	52,35	20
Term-13	2.515	93,14	20
Term-14	2.851	97,58	20
Term-15	9.854	58,19	20
Term-16	10.905	52,56	20
Term-17	1.426	103,29	20
Term-18	3.738	81,15	20
Term-19	62	9.672,37	20
Term-20	8.823	64,71	20
Term-21	8.859	65,34	20
Term-22	2.713	77,99	20
Term-23	10.701	53,82	20
Term-24	2.893	76,92	20
Term-25	10.960	52,49	20
Term-26	9.942	57,46	20
Term-27	5.422	48,72	20
Term-28	3.092	91,96	20
Term-29	65	9.234,85	20
Term-30	69	8.684,10	20
Term-31	5.550	66,29	20
Term-32	2.772	98,67	20
Term-33	9.826	58,83	20
Term-34	8.975	64,42	20
Term-35	8.473	68,22	20
Term-36	10.290	55,29	20
Term-37	3.487	76,03	20
Term-38	2.271	96,63	20
Term-39	9.918	58,16	20
Term-40	9.625	59,95	20
Term-41	3.256	79,44	20
Term-42	67	8.948,45	20
Term-43	24	155,63	20
Term-44	9.993	57,78	20
Term-45	11.047	52,74	20
Term-46	2.861	91,49	20
Term-47	205	165,78	20
Term-48	3.416	76,66	20
Term-49	534	210,94	20
Term-50	69	8.685,48	20

SERVIDOR 2

Terminal	Quantidade Transações	Média TPMc (ms)	Duração (min)
Term-01	2.407	80,47	20
Term-02	6.336	71,83	20
Term-03	4.458	102,05	20
Term-04	122	3.786,35	20
Term-05	109	4.232,31	20
Term-06	5.588	81,62	20
Term-07	5.169	88,24	20
Term-08	6.215	73,16	20
Term-09	5.115	89,10	20
Term-10	5.187	87,82	20
Term-11	116	3.975,26	20
Term-12	4.888	93,05	20
Term-13	104	4.440,07	20
Term-14	193	231,23	20
Term-15	350	143,03	20
Term-16	5.028	90,61	20
Term-17	5.820	78,43	20
Term-18	3.873	117,72	20
Term-19	908	153,23	20
Term-20	5.938	76,66	20
Term-21	2.948	121,51	20
Term-22	6.938	65,81	20
Term-23	3.660	124,17	20
Term-24	4.420	102,91	20
Term-25	5.032	90,75	20
Term-26	5.263	86,56	20
Term-27	5.967	76,25	20
Term-28	953	87,46	20
Term-29	5.082	89,69	20
Term-30	5.939	76,70	20
Term-31	4.614	98,76	20
Term-32	4.536	100,62	20
Term-33	360	190,44	20
Term-34	5.304	85,98	20
Term-35	105	4.400,87	20
Term-36	6.636	68,79	20
Term-37	5.213	87,22	20
Term-38	4.749	95,99	20
Term-39	4.907	84,91	20
Term-40	5.235	87,05	20
Term-41	905	114,84	20
Term-42	5.075	89,63	20
Term-43	5.524	82,61	20
Term-44	6.843	66,59	20
Term-45	4.910	92,63	20
Term-46	6.596	69,07	20
Term-47	4.661	97,91	20
Term-48	6.225	73,33	20
Term-49	4.675	97,40	20
Term-50	4.717	88,39	20

SERVIDOR 3

Terminal	Quantidade Transações	Média TPMc (ms)	Duração (min)
Term-01	26.589	21,97	20
Term-02	21.269	25,37	20
Term-03	2.607	97,58	20
Term-04	130	219,83	20
Term-05	1.012	129,26	20
Term-06	110	346,18	20
Term-07	5.120	69,10	20
Term-08	940	107,91	20
Term-09	4.713	68,48	20
Term-10	5.181	70,80	20
Term-11	551	68,99	20
Term-12	4.050	72,72	20
Term-13	6.342	54,55	20
Term-14	116	5.169,71	20
Term-15	27.660	21,09	20
Term-16	1.182	111,54	20
Term-17	693	109,71	20
Term-18	979	104,32	20
Term-19	126	195,07	20
Term-20	3.062	75,81	20
Term-21	2.327	84,25	20
Term-22	6.179	58,98	20
Term-23	20.508	26,83	20
Term-24	3.578	66,24	20
Term-25	404	128,83	20
Term-26	720	114,60	20
Term-27	2.254	76,77	20
Term-28	3.202	93,21	20
Term-29	1.496	88,93	20
Term-30	28.213	20,76	20
Term-31	1.091	90,20	20
Term-32	2.708	92,64	20
Term-33	677	118,58	20
Term-34	5.115	57,83	20
Term-35	2.173	99,11	20
Term-36	6.995	57,31	20
Term-37	6.022	56,07	20
Term-38	6.221	59,17	20
Term-39	15.281	31,36	20
Term-40	7.939	50,30	20
Term-41	127	4.724,16	20
Term-42	122	4.916,50	20
Term-43	7.710	51,23	20
Term-44	4.320	75,22	20
Term-45	402	113,22	20
Term-46	5.711	63,45	20
Term-47	2.340	83,70	20
Term-48	4.634	67,23	20
Term-49	7.516	50,13	20
Term-50	4.130	72,53	20