

UNIVERSIDADE DO EXTREMO SUL CATARINENSE – UNESC

CURSO DE CIÊNCIA DA COMPUTAÇÃO

SAMUEL LUIZ GOUDINHO

**AVALIAÇÃO DE DESEMPENHO NO ACESSO A BASE DE DADOS DE
IMAGENS ENTRE SGBDS POSTGRESQL E INFORMIX**

CRICIÚMA, DEZEMBRO DE 2011

SAMUEL LUIZ GOUDINHO

**AVALIAÇÃO DE DESEMPENHO NO ACESSO A BASE DE DADOS DE
IMAGENS ENTRE SGBDs POSTGRESQL E INFORMIX**

Trabalho de Conclusão de Curso apresentado
para obtenção do Grau de Bacharel em Ciência
da Computação da Universidade do Extremo
Sul Catarinense.

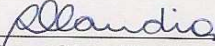
Orientador: Prof. MSc. Paulo João Martins

CRICIÚMA, DEZEMBRO DE 2011

SAMUEL LUIZ GOUDINHO

**AVALIAÇÃO DE DESEMPENHO NO ACESSO A BASE DE DADOS
DE IMAGENS ENTRE SGBDS POSTGRESQL E INFORMIX**

Submetido ao corpo docente do Curso de Ciência da Computação da Universidade do Extremo Sul Catarinense como um dos requisitos para obtenção do grau de Bacharel em Ciência da Computação.



Profa. MSc. Ana Claudia Garcia Barbosa
Coordenadora do Curso de Ciência da Computação

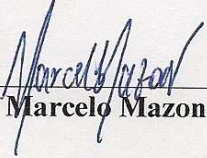
Banca Examinadora:



Prof. MSc. Paulo João Martins (UNESC)
Orientador



Prof. Esp. Fabricio Giordani (UNESC)



Esp. Marcelo Mazon (TI UNESC)

*Dedico essa pesquisa
a todos que tem forças para alcançar seus objetivos
em um mundo que nada é tão ruim
que não possa piorar.*

AGRADECIMENTOS

Agradeço primeiramente a Deus, onde sem ele, nada somos.

Agradeço também a minha mãe Jucélia Goudinho, pessoa paciente, que está sempre ao meu lado me apoiando psicologicamente e às vezes financeiramente.

A Luana Búrigo Medeiros que não teve paciência e me botou na parede, onde o nosso namoro estava condicionado ao término desse trabalho.

A UNESCO que me deu a oportunidade e as instalações necessárias para concluir mais um passo importante da minha vida profissional.

A todos os professores que ao longo do curso passaram seus conhecimentos, em especial ao meu orientado e amigo, Sr Paulo João Martins, que não mediu esforços para me auxiliar no desenvolvimento desse trabalho.

Em geral a todos, em especial ao meu amigo Gaudério, que sempre me incentivaram e acreditaram em mim.

*“ O único lugar em que sucesso vem
antes do trabalho é no dicionário ”*

Albert Einstein

RESUMO

A informação rápida e precisa é fundamental para a sustentabilidade de empresas e organizações em um mundo tão competitivo. Com isso, o presente trabalho, realizou uma comparação de desempenho entre os Sistemas Gerenciadores de Bancos de Dados PostgreSQL e Informix. Para a realização dos testes, foram utilizados arquivos de imagens, como carga de trabalho, e a ferramenta Benchmarksql, em ambiente GNU/Linux para medir o desempenho. Para obter resultados confiáveis, antes de começar a executar as análises, foi necessário realizar alguns testes aleatórios para calcular o número mínimo de amostras. No decorrer das análises, a cada 10 testes realizados a carga de trabalho sofria um acréscimo gradativo. Como resultado, apesar de o Informix sofrer a menor queda de rendimento na medida em que se aumentava o tamanho da carga, o PostgreSQL obteve um desempenho final superior aquele.

Palavras-chaves: PostgreSQL; Informix; Imagens; *Benchmark*.

ABSTRACT

The fast and accurate information is critical to the sustainability of companies and organizations in such a competitive world. Thus, the present work, realized a performance comparison between the Management Databases Systems PostgreSQL and Informix. For the tests, were used image files, as workload, and the tool Benchmarksql, on GNU/Linux to measure performance. Obtaining reliable results, before starting to perform the analysis, it was necessary to make some random tests to calculate the minimum number of samples. During the analysis, every 10 tests the workload suffered a gradual increase. As a result, although the Informix suffer the lowest drop in performance according it was increased the size of the load, PostgreSQL obtained a superior final performance than that.

Keywords: PostgreSQL; Informix; Images; Benchmark.

LISTA DE ILUSTRAÇÕES

Figura 1. Metadados de um objeto imagem.....	30
Figura 2. Localização do pixel $f(x,y)$	32
Figura 3. Elementos de um sistema de processamento de imagens.....	33
Figura 4. Interface gráfica do Benchmarksql	47
Figura 5. Nível de confiança de 95%	55
Figura 6. Gráfico comparativo do desempenho com uma imagem	60
Figura 7. Gráfico comparativo do desempenho com carga de 8 imagens	60
Figura 8. Gráfico comparativo do desempenho com carga de 16 imagens	61
Figura 9. Gráfico comparativo do desempenho com carga de 32 imagens	62
Figura 10. Gráfico comparativo do desempenho com carga de 64 imagens	62
Figura 11. Comparação do desempenho em cargas de imagens com tamanho de 582KB.....	63
Figura 12. Comparação do desempenho em cargas de imagens com tamanho de 1MB	64
Figura 13. Comparação do desempenho do Informix X PostgreSQL	64

LISTA DE TABELAS

Tabela 1. Configuração do Benchmarksql.	52
Tabela 2. Resultados de testes aleatórios	53
Tabela 3. Resultados das análises com imagens de 582KB no Informix	57
Tabela 4. Resultados das análises com imagens de 1MB no Informix.....	57
Tabela 5. Resultados das análises com imagens de 582KB no PostgreSQL.....	58
Tabela 6. Resultados das análises com imagens de 1MB no PostgreSQL	59

LISTA DE SIGLAS

ARO	Army Research Office
AS ³ AP	Ansi SQL Standard Scalable and Portable
BMP	Bitmap Image File
BSD	Berkeley Software Distribution
DARPA	Defense Advanced Research Projects Agency
GIF	Graphics Interchange Format
I/O	Input/Output
JDBC	Java Database Connectivity
JPEG	Join Photograph Experts
MO	Magneto-Óptico
NFS	National Science Foundation
OLTP	On-Line Transaction Processing
OR	Objeto Relacional
OSDB	Open Source Data Base
RAM	Random Access Memory
SGBD	Sistema Gerenciador de Banco de Dados
SGBDOO	Sistema Gerenciador de Banco de Dados Orientado a Objeto
SGBDR	Sistema Gerenciador de Banco de Dados Relacional.
SIG	Sistemas de Informações Geográficas
SO	Sistema Operacional
SQL	Structured Query Language
TIFF	Tagged Image File Format
TPC	Transaction Processing Performance Council

WORM

Write One Read Many

SUMÁRIO

1 INTRODUÇÃO.....	15
1.1 OBJETIVO GERAL	17
1.2 OBJETIVOS ESPECÍFICOS	17
1.3 JUSTIFICATIVA	17
1.4 ESTRUTURA DO TRABALHO	19
2 BANCO DE DADOS E SISTEMA GERENCIADOR DE BANCO DE DADOS.....	21
2.1 INFORMIX.....	25
2.2 POSTGRESQL	26
3 BASE DE DADOS PARA IMAGENS	29
3.1 IMAGENS DIGITAIS.....	31
3.2 SISTEMA DE PROCESSAMENTO DE IMAGENS	33
3.2.1 Aquisição.....	34
3.2.2 Armazenamento	34
3.2.3 Processamento	35
3.2.4 Transmissão.....	35
3.2.5 Exibição.....	36
4 AVALIAÇÃO DE DESEMPENHO EM SGBD	37
4.1 A TECNICA DE BENCHMARK	40
4.2 CARGA DE TRABALHO	44
4.3 OSDB.....	44
4.4 BENCHMARKSQL	46
5 TRABALHOS CORRELATOS	49

6 ANÁLISE COMPARATIVA DE DESEMPENHO ENTRE SGBDS OR NO	
ACESSO A BASE DE DADOS DE IMAGENS.....	50
6.1 PROCEDIMENTO METODOLÓGICO.....	50
6.2 TESTES.....	53
6.3 RESULTADOS.....	59
CONCLUSÃO.....	66
REFERÊNCIAS.....	68
APÊNDICES.....	71
ANEXOS.....	82

1 INTRODUÇÃO

Com o passar do tempo, a informação vem se tornando cada vez mais um ponto determinante para a tomada de decisão em empresas e organizações. Em um mundo tão competitivo, onde grandes valores são negociados em minutos, uma informação rápida e precisa pode ser determinante. Para tal, grandes bancos de dados são mantidos e vem a refletir as informações reais sobre a situação atual do negocio. Um banco de dados esta relacionado a um conjunto de informações, seja ela qual for, armazenado em uma base de dados. Informações estas que relacionadas de uma forma lógica, representam aspectos do mundo real. Para que esses objetivos sejam alcançados, é necessária a utilização de um conjunto de sistemas que possibilitem tanto a criação quanto o gerenciamento e manipulação do banco de dados, o que chamamos de SGBD (Sistema Gerenciador de Banco de Dados).

Esse conjunto de sistemas encontra-se localizada entre o banco de dados e os usuários do sistema, assim, é ele quem gerencia e coordena todas as interações entre usuário e sistema. Segurança, restrições de acesso, busca e atualização de arquivos são todos recursos do próprio SGBD utilizado. Outra característica relevante é que esse gerenciamento tenta isolar o usuário do sistema a níveis de hardware, disponibilizando uma visão acima, possibilitando que usuários interajam com o banco através de operações SQL.

O SGBD não é o único componente de software necessário para o funcionamento de um sistema, porem é de longe o mais importante (DATE, 2000). Como a questão de integridade, segurança e acesso da informação são de responsabilidade do próprio SGBD e não do desenvolvedor do sistema ou do usuário, surge à preocupação com a questão de seu próprio desempenho.

Os SGBDs surgiram para suprir as necessidades antes encontradas nos programas de processamento de arquivos. Uma das maiores características é o fato de apresentar um ambiente tanto conveniente quanto eficiente para a manipulação dos dados armazenados.

Com base nas referências bibliográficas utilizadas, fica evidente que com a complexidade dos novos sistemas, a informação evoluiu de simples arquivos de textos para arquivos com caráter multimídia, sons, vídeos, imagens. Com isso, surge a necessidade de se obter um SGBD que obtenha um eficiente desempenho quando incorporado a sistemas que utilizem imagens como base de dados. Como exemplo sistemas na área médica, onde exames e laudos médicos começam a ser armazenados como arquivos de imagens.

Para a realização de avaliação de desempenho, tanto em bancos de dados quanto a outros dispositivos computacionais, seja hardware ou software, é utilizada uma técnica conhecida como *benchmark*. Especialista em testes de desempenho, essa ferramenta possibilita a definição e execução de operações que simulem o funcionamento de um produto para assim obter valores relacionados ao seu funcionamento. Operações essas definidas com um volume de informações considerável, gerando o chamado teste de estresses, o que torna possível a análise do desempenho.

A partir das informações acima citadas, essa pesquisa tem como objetivo a realização de uma avaliação de desempenho entre os SGBD PostgreSQL e Informix.. Para a realização dos testes , serão utilizadas imagens como fonte de informação para o banco de dados. O teste serão executados por meio de uma ferramenta *benchmark*, uma ferramenta especialista que tem por objetivo obter valores que representam o quando

um produto é eficiente quando submetido a determinadas situações. O que geral valores passíveis a uma comparação.

1.1 OBJETIVO GERAL

Comparar o desempenho de SGBDOR no acesso à base de dados de imagens por meio de uma ferramenta *benchmark*.

1.2 OBJETIVOS ESPECÍFICOS

Os objetivos específicos relacionados com essa pesquisa são:

- a) descrever e aplicar os conceitos de banco de dados orientado a objetos e objeto relacional;
- b) documentar modelos existentes e fazer uso de uma ferramenta *benchmark* para avaliação de SGBD;
- c) definir os estudos de casos a serem aplicados a análise de desempenho;
- d) estudar e aplicar as métricas de comparação entre SGBDs.

1.3 JUSTIFICATIVA

Hoje em dia, Sistemas Gerenciadores de Banco de Dados Relacional são de longe os mais utilizados comercialmente. Mesmo obtendo resultados eficientes nas mais diversas áreas, esse modelo possui algumas limitações quando incorporados a sistemas complexos ou não convencionais.

Com o advento da programação orientada a objeto, surge necessidade da utilização de um banco de dados que venha a dar um suporte maior ao que diz respeito a essa metodologia. O SGBDOO agrupa recursos de banco de dados com os conceitos de orientação a objetos, como persistência, encapsulamento, polimorfismo e hierarquia, dando suporte para que os objetos representem aspectos do mundo real. Apesar de possuírem poderosos mecanismos de manipulação de objeto, Bittencourt (2007), ressalta que esse modelo de banco de dados ainda não possui um grau de maturidade aceitável para que seja responsável pelo gerenciamento de grandes volumes de dados e operações.

Devido a esse grau de imaturidade, o modelo de SGBD objeto relacional veio a suprir a necessidade de se trabalhar com orientação a objetos e agregar características do modelo relacional, como a utilização da linguagem de consulta SQL. Para tal evolução, esse modelo se apóia em características como versatilidade, robustez e outras tecnologias do modelo relacional, com a flexibilidade e potencialidade tecnologia orientada a objeto (BITTENCOURT, 2007).

Como falado no tópico anterior, a informação vem se tornando essencial para a competitividade de organizações em um mundo tão concorrido. Devido ao seu potencial, a informação vem deixando de ser apenas texto e incorporando um caráter multimídia, imagens, sons, vídeos. Uma tendência em ascensão em sistemas no ramo da medicina, arqueologia, cartografia, meteorologia, entre outros (BITTENCOURT, 2007). Não muito distante da realidade, a Universidade do Extremo Sul Catarinense (UNESC), proporciona um contínuo desenvolvimento de aplicações na área da saúde, onde muitos desses sistemas se apóiam a utilização de imagens, para a obtenção de laudos, diagnósticos, entre outros.

Com o levantamento bibliográfico realizado, nota-se uma carência de documentação de pesquisas na área de avaliação de desempenho entre SGBDOR. Oposto a isso é a utilização de banco de dados relacional, principalmente utilizando textos como fonte de dados. E com base nessa carência e na importância do assunto, sente-se a necessidade tanto de avaliar, quanto a de documentar uma avaliação de desempenho entre SGBDs que utilizem imagens como fonte de dados.

Para que seja possível essa avaliação, uma ferramenta *benchmark* específica para testes em banco de dados será utilizado. Essa ferramenta realiza uma carga de operações suficiente para a realização de uma análise de desempenho. Voltados a análise de SGBDs, podemos citar algumas ferramentas *benchmark* como o OO1 Benchmark, HyperModel Benchmark, OO7 Benchmark, OSDB (Open Source Data Base), Benchmarksql, entre outros.

Um dos pontos a se alcançar com essa pesquisa é a de minimizar os impactos causados pela escolha incerta ou equivocada de um SGBD. Onde os resultados contidos neste trabalho poderão servir de guia para futuras escolhas.

1.4 ESTRUTURA DO TRABALHO

O trabalho aqui realizado se apresenta em uma estrutura com oito capítulos na seguinte forma:

Uma breve introdução ao trabalho a ser realizado se encontra logo no primeiro capítulo. Apresentando também uma definição geral do problema a ser solucionado, objetivos específicos para se alcançar a solução e uma justificativa para se estar realizando tal trabalho.

A partir desses conceitos, o capítulo dois faz referência a modelos de SGBD, com enfoque no modelo relacional, orientado a objeto e objeto relacional, modelos esses mais facilmente encontrados. O capítulo se encerra com uma descrição sobre as duas ferramentas de banco de dados que serão utilizadas nesse trabalho, o PostgreSQL e o Informix.

Como fonte de dados para os bancos, as imagens são englobadas no terceiro capítulo, apresentando características e conceitos que estão relacionados a bases de dados para imagens. Apresenta também conceitos sobre imagens digitais e os passos de um sistema de processamento de imagens.

O quarto capítulo apresenta uma série de conceitos relacionados com a prática da avaliação de desempenho de SGBD e ferramentas que auxiliam de forma eficaz nessa avaliação. Finalizando, é apresentada a técnica de *benchmark*, algumas características e uma descrição de algumas ferramentas, bem como as utilizadas nos testes desse trabalho, o Benchmarksql. Algumas sugestões de futuros trabalhos são propostas no capítulo cinco.

Todas as etapas referentes à parte prática podem ser encontradas no sexto capítulo, descrevendo o ambiente utilizado, a configuração utilizada no *benchmark*, métricas e cálculos de amostra, bem como todos os testes, resultados obtidos e comparações. Finalizando, o capítulo sete se refere à conclusão e o capítulo oito apresenta as referências bibliográficas utilizadas para a realização deste trabalho.

2 BANCO DE DADOS E SISTEMAS GERENCIADOR DE BANCO DE DADOS

A informação rápida e precisa possui um papel fundamental para empresas e organizações. A informação é gerada pelo relacionamento de diferentes dados relativos às organizações, armazenados em um repositório de banco de dados. Uma coleção de dados que, quando relacionados a outras, representam aspectos sobre um objetivo específico.

Um Sistema Gerenciador de Banco de Dados é um conjunto de programas que possibilita tanto a criação quanto a contínua manutenção de um banco de dados. Permanecendo entre o banco de dados e a aplicação, ele tenta isolar ao máximo o usuário em nível de hardware e ao mesmo tempo tenta proporcionar um ambiente tanto conveniente quanto eficiente para manipulações do banco de dados (SILBERSCHATZ; KORTH; SUDARSHAN, 1999).

Os SGBDS vieram a substituir e suprir as necessidades e desvantagens antes encontradas com a utilização dos programas de processamento de arquivos, onde eram armazenadas as informações. Para tal feito, ele se apóia em algumas funções fundamentais como, métodos de acesso ao banco de dados, controle de redundâncias, segurança contra falhas e contra acesso não permitido de usuários, integridade semântica para garantir que os dados são armazenados corretamente e concorrência para evitar conflitos de acesso simultâneo.

Da sua criação, até os tempos de hoje, os bancos de dados sofreram enormes modificações. Essas evoluções foram diferenciadas em primeira, segunda e terceira geração. Os bancos de dados de primeira geração se caracterizam pelo poder de distinção do que era informação lógica e do que era informação física. Para isso, modelos de dados começaram a ser utilizados para definir uma estrutura física dos

dados de uma forma lógica. Os modelos utilizados foram o modelo em rede e o modelo hierárquico. O acesso ao banco era feito através de ponteiro de baixo nível, o que obrigava os usuários a conhecer suas estruturas para poder manipulá-lo.

Em meados das décadas de 70 e 80, Edgar Frank Codd propõe um modelo que tirava a ligação entre a estrutura lógica do banco de dados do método de armazenamento físico. A segunda geração começa a surgir a partir então do modelo de dados relacional.

O SGBD Relacional trabalha com o armazenamento dos dados em tabelas, cada qual com uma identificação única. Os valores de uma tabela se relacionam através de ligações com valor de outras tabelas, fazendo assim o relacionamento entre os dados e transformando dados em informação. Esse modelo é de longe o mais utilizado na atualidade em aplicações tradicionais, onde em alguns casos os dados armazenados estão restritos a números e textos.

Com o passar dos tempos, novas aplicações de banco de dados foram surgindo, como Banco de Dados Hipertexto e Banco de Dados Multimídia, onde novos arquivos são armazenados como imagens, sons, vídeos. Essas aplicações, consideradas não convencionais, estão sendo prejudicadas por limitações do modelo relacional. Isso porque esses e muitos outros tipos de arquivos hoje armazenados não eram considerados quando esse e outros modelos mais antigos foram criados (SILBERSCHATZ; KORTH; SUDARSHAN, 1999).

Tentando minimizar os impactos sofridos por novos sistemas em virtude de modelos de SGBD inadequados, novos modelos estão sendo projetados e utilizados na atualidade, como por exemplo, o modelo orientado a objeto. Alguns dos processos necessários para suprir as necessidades dos SGBDs já existiam a tempos dentro da linguagem de programação orientada a objeto, o que alavancou a criação de um novo

modelo. Assim, a terceira geração veio a dar uma integração maior entre o banco de dados e a linguagem de programação das aplicações.

O desenvolvimento do Sistema Gerenciamento Orientado a Objetos tomou como fundamento os conceitos dos bancos de dados tradicionais em junção com os recursos dos conceitos de programação orientada a objetos. Neste modelo, cada informação é armazenada na forma de objetos. Cada objeto é composto pelas variáveis que são os dados a serem armazenados e uma série de métodos que operam sobre esse objeto. Assim, possibilita a utilização de novos modelos de dados, considerados complexos ou não convencionais como números e textos.

Esse modelo de banco de dados possui algumas propriedades que ajudam a entender melhor o seu funcionamento, como:

- a) **persistência transparente:** com a finalidade de diminuir a complexidade do código da aplicação, essa propriedade possibilita manipular os dados armazenados no banco através da própria linguagem orientada a objetos, diferente do banco de dados relacionais que utilizam uma sub-linguagem, a SQL;
- b) **ausência de impedância mismatch:** impedance mismatch é a denominação dada ao processo em que o banco de dados necessita mapear todos os objetos da aplicação para o modelo relacional, o que já não acontece com SGBDOO, onde os objetos são armazenados diretamente na bases, possibilitando que o mesmo modelo de classe utilizado na aplicação seja também utilizado no banco de dados;
- c) **identidade do objeto:** os objetos em um banco de dados orientados a objetos possui uma identidade imutável, ou seja, não sofre alteração mesmo com a alteração de todos os valores de seus atributos. O mesmo

não ocorre em banco de dados relacional, onde os valores de uma tupla são identificados através de um atributo chave.

Esses objetos estão estruturados em uma hierarquia de tipos e subtipos, onde assim herdam recursos de seus superiores. Do paradigma de orientação a objetos, esse modelo possui algumas características e funcionalidades bem conhecidas, como herança, polimorfismo e encapsulamento.

A utilização desse modelo, para, Elmasri e Natvthe (2002), se dá por uma série de fatores como a crescente utilização de linguagens de programação orientada a objetos, uma realidade em plena ascensão. Bem como pelo fato de possibilitarem que desenvolvedores definam as estruturas dos objetos especificando quais operações podem ser aplicadas a esses objetos. Porém esses modelos de dados ainda não apresentam uma maturidade suficiente para serem incorporados a grandes volumes de dados (SILBERSCHATZ; KORTH; SUDARSHAN, 1999).

O modelo objeto relacional veio a unir o útil com agradável. Com a junção das características do modelo relacional com a orientação a objetos, esse modelo vem como uma evolução do modelo relacional. Para tal evolução, esse modelo se apóia em características como versatilidade, robustez e outras tecnologias do modelo relacional, com a flexibilidade e potencialidade tecnologia orientada a objeto (BITTENCOURT, 2007).

Essas características em um só modelo deram o suporte para serem utilizadas com sucesso nas mais diversas aplicações, o que fez com que, muitas empresas desenvolvedoras incorporassem esses modelos em seus produtos.

Em ambos os casos, tanto no banco de dados orientado a objeto quanto no banco de dados objeto-relacional, o paradigma de orientação a objeto se faz presente em sua implementação, se diferenciando apenas na sua manipulação.

2.1 INFORMIX

O Informix é um SGBDOR projetado e desenvolvido por Roger Sippl no final dos anos 70. A companhia foi fundada nos anos 80, tornando o seu produto publico em meados de 1986.

Após algumas versões não bem sucedidas, em 1999, foi lançada uma nova versão, a 7.31, tendo um código fonte totalmente reescrito, o que trouxe ao novo produto uma confiabilidade antes não alcançada, bem como algumas outras funções que o fizeram ser considerado, o segundo banco de dados da época, atrás apenas do poderoso Oracle. Em 2001, por sugestão do mercado Wal-Mart e sabendo de uma possível falência da Informix, a IBM fez a aquisição desse produto. Unindo-o com o banco de dados DB2, já desenvolvido pela própria empresa, o produto se tornou uma ferramenta que competia diretamente com o Oracle.

Com o passar do tempo, os investimentos em desenvolvimento e tecnologias aplicados ao banco foram diminuindo, a um ponto de darem um ponto final ao projeto, destinando todos os seus esforços apenas ao DB2. Com isso, notou-se que muitos clientes que o utilizavam não migraram para o DB2, e sim para seus concorrentes, o que despertou o seu desenvolvimento em 2004.

O retorno do investimento em pesquisas e desenvolvimento do Informix , fez com que a IBM o colocasse como o banco de dados OLTP oficial e o DB2 como uma solução para Mainframes (DOES, 2011). Novas versões foram sendo desenvolvidas como a versão 10 do Informix IDS em 2005. A versão 11 foi lançada em meados de 2007, tendo hoje uma pesquisa direcionada a desenvolvimento para utilização em plataformas para soluções embarcadas, competindo assim com produtos

como o MYSQL. Uma nova versão, a de numero 12, permanece em desenvolvimento, sendo esperado seu lançamento para meados de 2012.

2.2 POSTGRESQL

O PostgreSQL é um SGBDOR, que encontra-se entre os mais famosos existentes no âmbito dos softwares livres, sendo considerado um dos fortes concorrente do Oracle.

Atualmente, é desenvolvido por pessoas e empresas de varias partes do mundo, onde é distribuído sob a licença *Berkeley Software Distribution* (BSD). Essa licença possibilita que qualquer usuário tem acesso ao código fonte da aplicação, podendo assim adequado a diferentes situações, se restringindo apenas a parte do código que se refere a direitos autorais e licenças licenciadas.

Seu projeto foi iniciado em 1986, pelo programador e professor Michael Stonebraker, na Universidade de *Berkeley*, Califórnia. Na época, a universidade já contava com uma aplicação semelhante, o *Ingres*, que serviu como base para a criação do projeto Postgres, um servidor de banco de dados relacional. Para tal, foi patrocinado por diversas organizações como a *Defense Advanced Research Projects Agency* (DARPA), o *Army Research Office* (ARO), a *National Science Foundation* (NFS) e pela ESL, Inc (PostgreSQL Global Development Group, 2007).

Como resultado do projeto, uma versão de demonstração do Postgres foi apresentada em 1987, denominada *The implementation of POSTGRES*, sendo liberada a versão 1.0, a primeira, apenas em junho de 1989. A partir dessa data, algumas outras versões foram sendo criadas para suprir limitações e necessidades do mercado, como:

- a) **versão 2.0:** com mudanças no sistemas de regras, ponto muito criticado na versão anterior, a versão 2.0 foi desenvolvida e lançada em 1990;
- b) **versão 3.0:** essa nova versão possibilitou suporte a múltiplos gerenciadores de armazenamento, trouxe melhorias no executor dos comandos e reescreveu o sistema de regras (PostgreSQL Global Development Group, 2007);
- c) **versão 4.2:** em 1992 a empresa Illustra Information Technologies adquiriu e comercializou o código do Postgres. Porém, no ano seguinte, a empresa observou que o número de usuários havia crescido, e que o alto tempo destinado a manutenção e suporte poderiam ser utilizados de uma forma mais satisfatória em estudos referentes a banco de dados. Em virtude, o desenvolvimento do Postgres foi encerrado com a versão 4.2.

Em 1994, Jolly Chen e Andrew Yu, estudantes de Berkeley, introduziram um interpretador de linguagem SQL ao projeto que passa a se chamar Postgre95. Muitas vantagens foram obtidas em relação a versão anterior, como a utilização de consultas SQL no lugar de *PostQUEL*, bem como mudanças internas que o deixaram com um melhor desempenho. Deixando de lado o rótulo de 95, em 1996, a versão 6.0 surge com um novo nome, PostgreSQL, fazendo uma referência a linguagem de consulta.

Atualmente, o PostgreSQL encontra-se na versão 8.4 e não deixa a desejar em relação a grandes SGBDs como a Oracle e a SQL Server. Segundo Colares (2007), é um SGBD flexível que pode tanto ser utilizado em pequenas aplicações, por ser gratuita, quando em aplicação robustas com operações de missão crítica. Com ele, o desenvolvedor tem como optar em utilizar diversas linguagens de programação, bem como diferentes plata

Com ele pode-se desenvolver aplicações nas mais variadas linguagens de programação como Java, Delphi, C/C++, bem como sob diferentes plataformas como Windows, Linux, Solares, entre outros. Mantendo a evolução dos tempos, a versão 9.0 já se encontra em pleno desenvolvimento.

3 BASE DE DADOS PARA IMAGENS

Com a crescente evolução dos SGBDs e das necessidades, os dados em um banco de dados, em muitas áreas, vem deixando de lado o caráter textual e incorporando arquivos como áudio, vídeo e imagens. Dentre os arquivos que compreendem o caráter multimídia, aos arquivos de imagens estão largamente à frente entre os mais utilizados. Essas bases de dados são cada vez mais comuns em aplicação como SIG, em áreas da biologia, cartografia, entre outros. Bem como também na área da medicina, que fazem utilidade de arquivos de imagens diariamente, armazenando laudos médicos, radiografias, tomografias entre outros. As imagens podem ser armazenadas em diferentes formatos, porém, sempre em resoluções suficientemente altas para que não ocorra perda de informação por meio de manipulações.

Esses bancos de dados de imagens devem ter o poder tanto de trabalhar com grandes volumes de informações, imagens, como também lidar com metadados, informações vinculadas a outras informações contidas no mundo digital. Esses dados podem estar ligados a autores da informação, data da criação, comentários e opiniões em geral, conforme a figura 1. Podendo ser manipulados, esses metadados fornecem tanto facilidades para a recuperação de informações digitais, como também nos dá o poder de entender, lidar e informar opiniões sobre elas (BITTENCOURT, 2007).



Figura1. Metadados de um objeto imagem.
Fonte: SILVA (2006 p11)

Para Gonzales (2003), algumas soluções tentam dar um suporte maior para os SGBDs lidarem com imagens, como por exemplo:

- a) **integração virtual**: uma solução simples, que não faz utilização de armazenamento do arquivo no banco. Isso porque, permanecem devidamente armazenados no banco apenas informações textuais que fazem referência ao arquivo, como nome, tamanho, comentários, manipulados fora da aplicação.
- b) **pseudo ou semi-integração**: nesta solução o armazenamento do arquivo multimídia no banco de dados se faz necessário, porém, a interpretação todas as manipulações sobre elas são feitas por meio de identificadores e é de responsabilidade da aplicação.
- c) **integração real**: aqui os arquivos são armazenados de forma agregada. O usuário tem todo o poder de manipular as informações usando funções do próprio SGBD, sem necessidade dos identificadores ou informação.

Segundo Gonzales (2003) para que o banco de dados forneça suporte para manipulação de arquivos de imagens, alguns requisitos, não contidos em banco convencionais também devem ser considerados, como:

- a) **algoritmos e estruturas de representação**: algoritmos e técnicas de processamento de imagens específicas para uma manipulação eficiente;
- b) **definição de critérios de similaridade**: a diferenciação de descritores por meio de critérios de similaridade;
- c) **expressão das imagens em linguagem de consulta**: *Query by Pictorial Exemple*, modelos de imagens representando o dado a ser consultado. Consultas textuais, possibilidade de consultar a imagem através de termos e conceitos. Consultas visuais, ferramentas gráficas são utilizadas para que o usuário possa definir características da imagem;
- d) **suporte para a indexação das imagens**: devido a complexidade da consulta de dados inexatos, a indexação de características das imagens deve fornecer uma acesso rápido e objetivo, como por exemplo, metadados;
- e) **integração de recuperação de dados tradicionais e imagens**: a linguagem de consulta deve fornecer suporte tanto para dados convencionais quanto para imagens.

3.1 IMAGENS DIGITAIS

Uma imagem pode ser representada de diversas formas no mundo real, que vão desde fotos com altas resoluções, pinturas de quadros até riscos em parede de cavernas. Para que essas imagens possam ser inseridas no mundo digital, elas são

transformadas e constituídas de pontos distribuídos em uma matriz de linhas e colunas (x, y) , conforme a figura 2. Cada ponto ou *pixel* se refere a uma cor específica definida por meio de um coeficiente $f(x, y)$ (CARVALHO, 2003).



Figura 2 .Ponto da matriz $f(x,y)$
Fonte: (CARVALHO, 2003)

Assim, pode se dizer que uma imagem é o resultado de uma matriz de pixels, onde cada índice de linhas e colunas determina um ponto da imagem, e o valor do elemento dessa matriz determina a cor, ou nível de cinza naquele ponto.

Dependendo da aplicação ou da finalidade, diferentes formatos de imagens podem ser adotados, como *Joint Photographic Expert (JPEG)* e *Microsoft Windows Bitmap (BMP)*, os mais conhecidos e utilizados, porém, não os únicos, existem ainda outros formatos como o GIF, TIFF, entre outros.

3.2 SISTEMA DE PROCESSAMENTO DE IMAGENS

Segundo Scuri (1999), as funções que compreendem o contexto de processamento de imagens podem ser divididas em duas classes, por escopo ou por resultado. Por escopo se refere às técnicas utilizadas no processamento da imagem, já por resultado são os resultados obtidos através da utilização dessas técnicas de processamento.

O processamento de imagens é constituído por alguns procedimentos, como aquisição, armazenamento, processamento e exibição, conforme mostra a figura3. Esses elementos, para Marque Filho e Vieira Neto (1999), são de uso genérico, podendo ser utilizados tanto em sistemas simples, quanto em sistemas sofisticados com uso intenso de imagem.

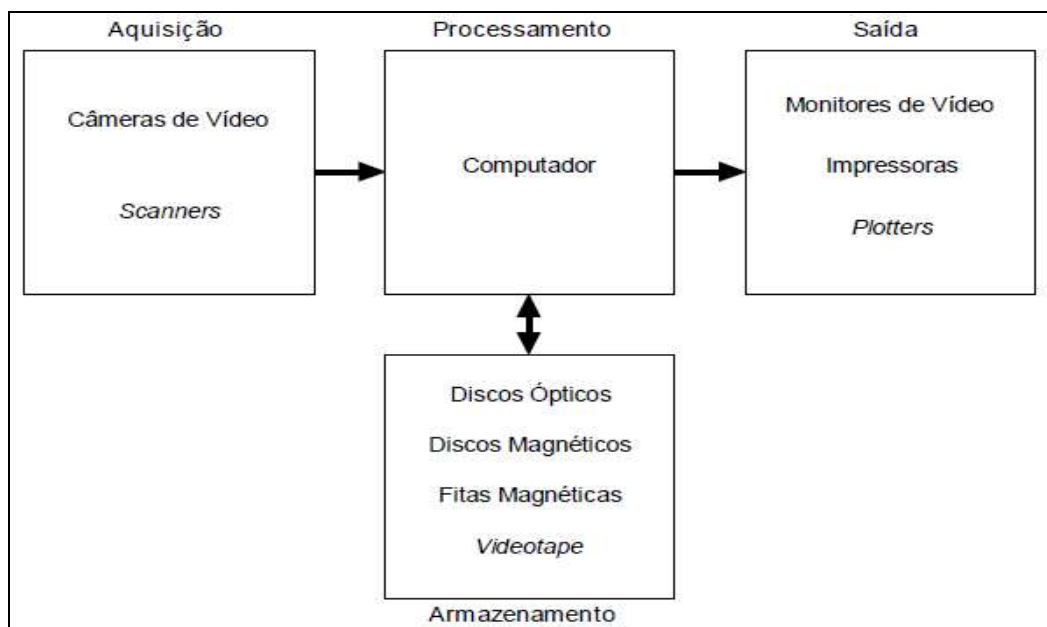


Figura3. Elementos de um sistema de processamento de imagens.
Fonte: Marque Filho e Vieira Neto (1999, p2)

3.2.1 Aquisição

O primeiro passo certamente se dá com a aquisição da imagem, ou seja, adquirir uma imagem digital. Esse elemento tem a função de transformar a imagem em um formato adequado, se necessário, para o processamento. Para isso, este processo se divide em duas etapas. A primeira está voltada a um sensor físico sensível a uma faixa de energia eletromagnética, responsável em produzir como saída um sinal elétrico equivalente ao nível de energia detectado. Na segunda etapa, um dispositivo chamado digitalizador, faz a conversão do sinal elétrico analógico para a forma digital.

3.2.2 Armazenamento

Segundo Mendes (2010), uma imagem com resolução de 1024 x 1024 pixels de oito bits resultam no armazenamento de aproximadamente um milhão de bits, o que torna esse armazenamento um dos grandes desafios nos projetos de processamento de imagens.

Assim, para Marque Filho e Vieira Neto (1999), o armazenamento de imagem pode ser dividido em três categorias:

- a) **armazenamento de curta duração de uma imagem:** uma opção é a utilização da própria memória do computador, memória RAM. A segunda opção faz utilização de cartões especialistas, também chamados de *frame buffers*, onde podem ser armazenadas quantas imagens forem necessários em alta velocidade. Tanta velocidade faz com que procedimentos de *zoom* (ampliação ou redução) e de *scroll* ou *pan* (rolagens) sejam executadas de forma instantâneas a olho nu;

- b) **arquivamento de massa para recuperação rápida de imagens:** também chamado de armazenamento on-line, esta categoria prioriza não a grandes capacidades de armazenamento, sim um rápido tempo de acesso aos dados. Para isso, faz o armazenamento em discos magnéticos, discos rígidos e em discos MO;
- c) **arquivamento para recuperações futuras:** esta categoria não prioriza o tempo de acesso, que pode ser aleatório em diferentes casos, mais sim a grande, ou melhor, gigantesca capacidade de armazenamento. Tamaña capacidade pode ser justificada pelo seu dispositivo de armazenamento, os discos ópticos WORM, com capacidade podem atingir aos 10 GB por disco.

3.2.3 Processamento

Como o processamento de imagens digitais são expressos geralmente sob a forma algorítmica, fora as etapas de aquisição e exibição, a maioria das funções de processamento de imagens pode ser implementadas via software. O uso de hardwares especialistas no processamento se faz necessário quando se encontram algumas limitações intoleráveis no computador que realizara o procedimento (MARQUES FILHO e VIEIRA NETO, 1999).

3.2.4 Transmissão

Como toda a transmissão, imagens podem ser transmitidas a distancia através de protocolos já existentes ou no caso, por uma rede de computadores. O

resultado desta etapa esta muitas vezes condicionada aos meios ou canais de comunicação, por onde grande quantidade de bits que se necessita transferirem por meio destes. Esse problema aumenta se considerarmos o envio de vídeos em tempo real, visto que o envio deve estar associado a uma sincronia necessária.

3.2.5 Exibição

Os monitores são os dispositivos mais utilizados para a exibição de uma imagem no formato digital. O monitor de cristal liquido ou LCD estão sendo muito utilizado devido à grande utilização de computadores portáteis, também ao baixo consumo de energia aliando a uma ótima qualidade na exibição da imagem. Também existe a possibilidade de exibição dessa imagem em papel, como a reprodução fotográfica e também em papel sensível a temperatura, cujo sua composição química faz realçar cores com o aumento da temperatura.

4 AVALIAÇÃO DE DESEMPENHO EM SGBD

Em qualquer aquisição, seja de um produto ou serviço, algumas características podem ser levadas em consideração para uma boa escolha, como preço, qualidade, desempenho, entre outros. Essa avaliação pode ocorrer dentro de um supermercado, na escolha de uma universidade, na compra de um carro, entre outros.

No ramo da ciência não é diferente, tanto produtos de hardware, softwares possuem características passíveis de comparação, encontradas a partir do que chamamos de análise de desempenho. Uma técnica que consiste em medir a eficiência sobre algumas determinadas funcionalidades, sobre algum determinado aspecto específico, considerando algum tipo de medida como tempo e distância (COSTERANO, 1997). No ramo da computação (SVOBODOVA,1976 appud DURÃES,2007) destaca que a técnica de avaliação de desempenho tem como objetivo verificar a efetividade com que os recursos de hardware são utilizados para atender aos objetivos do software. Onde essa avaliação pode ser comparativa, quando se compara desempenhos de sistemas diferentes, ou analítica, quando a comparação é realizada mediante parâmetros e configurações do próprio sistema.

Se tratando de avaliação de desempenho em SGBD, inúmeros fatores podem vir a interferir no seu desempenho, como o sistema computacional utilizado, hardwares e softwares, como o próprio desenvolvimento tanto do banco de dados e do sistema que o utilizará. Para, Coterano (1997, p.33), tanto o sistema quanto os usuários podem interferir diretamente no desempenho de um SBGD, definindo que:

O desempenho do usuário se refere a aspectos relativos à facilidade de utilização, nível de interface do usuário, aprendizagem da linguagem de programação, entre outros. O desempenho do sistema se preocupa basicamente com a avaliação de fatores quantitativos que procuram expressar a eficiência do sistema no atendimento de requisições dos usuários, tais como: o tempo de resposta, o *throughput*, número de referências lógicas e físicas a memória secundária.

Além disso, Costerano (1997), afirma que existem fatores que interferem no desempenho do SGBD e que os usuários não podem fazer nada, como o próprio código que o mesmo foi implementado. Porém, destaca também que algumas modificações de parâmetros, criação de índices e agrupamentos podem ser realizados com o intuito de otimizar a utilização dos recursos.

Na atividade da análise de desempenho, vários autores como Ciferri (1995), Costerano (1997) e Pereira (1990), destacam a utilização de três modelos que dever servir como fundamento:

- a) **modelo analítico**: este modelo leva como base a utilização de um conjunto de função ou equações matemáticas que, aliadas a algoritmos que os resolvem e relacionam medidas de desempenho de um SGBD passíveis de comparação. Para Pereira (1990), este modelo tenda a resultados menos precisos em virtude da utilização de registros de bancos e cargas de trabalho hipotéticas. Tendo assim um custo relativamente baixo, devido a facilidade da utilização e a rapidez dos resultados;
- b) **modelo de simulação**: este modelo procura reproduzir as atividades de um SGBD a partir de um conjunto de hipóteses mais prováveis, eliminando a análise do próprio. A simulação é realizada com base a entrada e saída (I/O) de dados como parte do mundo real. Este modelo, como o analítico, tende a resultados menos precisos, visto a dificuldade de se definir os parâmetros de um sistema muitas vezes em estágios primários de desenvolvimento (COSTERANO, 1997);
- c) **modelo experimental**: eliminando com valores em hipóteses, este modelo, também chamado de modelo empírico, utiliza o próprio SGBD

para a obtenção dos resultados de desempenho. Como isso, aumenta a precisão dos resultados adquiridos. Este modelo se baseia em duas técnicas, a de monitoramento e a de *benchmark*. A técnica de *benchmark* consiste na execução de uma carga de trabalho sobre o próprio SGBD. Para Costerano (1997), esta técnica pode ser utilizada para comparação de diferentes sistemas por ser padronizados, contendo testes invariáveis e bem definidos. Já a técnica de monitoramento consiste na utilização de ferramentas específicas de avaliação estatística contida no próprio sistema a ser analisado, tornando esta técnica muito limitada. Assim, a precisão dos resultados neste modelo depende muito da maturidade das tecnologias envolvidas, como exemplo as ferramentas *benchmark*.

Em alguns casos específicos, corre a utilização de modelos híbridos, a utilização de características e vantagens de dois ou mais modelos com a finalidade de uma representação mais precisa e realista do sistema.

Para Ciferri (1995) e para Costerano (1997) essa análise de desempenho pode tornar real algo passível de comparação, podendo responder questões como:

- a) **avaliação da capacidade máxima:** o objetivo geral da análise de desempenho é justamente responder o quanto um produto é eficiente quando submetido a uma determinada funcionalidade. Objetivo geral porém não o único, visto que o tempo não é a única medida de desempenho;
- b) **comparação entre tecnologias:** neste caso, os resultados obtidos com a análise de desempenho ajudam a identificação de restrições de desempenho perante produtos com tecnologias diferentes, mostrando

assim algumas reestruturações necessária. Para Costerano (1997), estas restrições, tanto em nível de pesquisa como a de tecnologias empregadas, tendem a mostrar os melhores caminhos para trabalhos futuros, como o que pode ser pesquisado, o que pode ser melhorado, visando que novos produtos se tornem comercializável;

- c) **avaliação da capacidade específica:** assim, pode analisar se uma determinado produto, seja ele hardware ou software, pode ser executado com um desempenho aceitável em alguma aplicação. Assim, pode-se avaliar uma aplicação qualquer se comportara de forma aceitável sobre uma arquitetura de hardware e software;
- d) **avaliação da relação custo x benefício:** essa é uma das formas mais coerentes do processo de seleção de um produto. Esta forma de avaliação permite selecionar os produtos que mais se aproximam tanto das necessidades da empresa e da aplicação, quanto à dos investimentos disponibilizados pela empresa para aquisição de tal produto. Onde, para empresas com diferentes orçamentos, diferentes produtos podem se mostrarem mais atraentes.

4.1 A TECNICA DE BENCHMARK

Em qualquer atividade, o tempo é índice indispensável para se alcançar a excelência. Como em uma corrida ou natação em uma competição, quem faz um trajeto específico em menor tempo se sobressai em relação aos competidores que não alcançaram suas marcas. Em empresas e organizações, setores buscam através de

estudos executarem processos em um tempo considerado bom, bem como o seu contínuo aprimoramento.

Surge então, no final dos anos 80, o termo *benchmarking*, que sugere um estudo estruturado de oportunidades de melhorias, onde com esse estudo mudanças são feitas para se obter um melhor desempenho. Com isso, o *benchmarking* apresenta-se como uma alternativa para se alcançar a excelência.

Para a realização de avaliação de desempenho, tanto em SGBD quanto a outros dispositivos computacionais, é utilizada uma técnica de *benchmark*. Uma ferramenta especializada em testes de desempenho, onde em SGBDs, possibilita a definição de uma série de transações, que quando executadas geram uma medida capaz de servir como base para comparações. Estas transações são chamadas de carga de trabalho, e a sua utilização é fundamental para uma análise justa e coerente. O fato de esta técnica ser executada em junção com próprio sistema a ser analisada dá o crédito necessário para uma análise muito próxima do real.

Existem diversas implementações de ferramentas *benchmark* disponíveis no mercado, como o AS³AP, OSDB, TPC, Wisconsin, Benchmarksql, entre outros. E para que eles obtenham sucesso, uma série de requisitos devem ser levados em consideração, como descreve (GRAY,1993 apud DURÃES,2007):

- a) **relevância**: que se refere a utilização de parâmetros reais da aplicação a ser avaliada pelo *benchmark*;
- b) **portabilidade**: o *benchmark* deve ser executado em diferentes arquiteturas computacionais e sob diferentes plataformas. O que possibilita a sua execução sob diferentes aplicações;

- c) **escalabilidade:** o *benchmark* pode ser executado tanto em pequenos como em super computadores, onde configurações e arquiteturas não devem influenciar na confiabilidade dos valores obtidos por ele;
- d) **simplicidade:** a ferramenta deve ser de fácil compreensão e execução.

Requisitos esses que devem em conjunto compreender uma ferramenta *benchmark* para que se obtenha a confiabilidade necessária para uma correta análise. É a partir dessas características de confiabilidade nos resultados que esse trabalho se dispôs a utilizar uma ferramenta de *benchmark* para realizar uma análise de desempenho em SGBDs.

4.2 CARGA DE TRABALHO

Conforme visto anteriormente, a técnica de *benchmark* possui uma portabilidade que possibilita a sua execução em análises nos mais variados ambientes e nos mais diversos produtos, seja software ou hardware. Se tratando de sistema de banco de dados, alguns *benchmarks* específicos para a análise dessas aplicações são facilmente encontrados.

Essas ferramentas possibilitam a definição de transações que posteriormente são executadas para medir o desempenho. Na análise de banco de dados, essas transações estão ligadas a operações de escrita e leitura de dados armazenados no banco de dados a ser analisado. Essas operações são geralmente implementadas em uma linguagem de programação, como o C, Java, Delphi, em conjunto com uma linguagem de programação geralmente disponibilizada pelo próprio SGBD. A essas transações ou sobrecargas de trabalho específicos, dá-se o nome de carga de trabalho.

É a sua execução por meio da ferramenta benchmark que o sistema será avaliado. Vale salientar que alguns processos podem ser executados para que essa sobrecarga de trabalho específica seja executada, o que também vem a degradar o sistema e a interferir de forma direta no desempenho do mesmo, é o que chamamos de carga de trabalho externa ao *benchmark*. A sobrecarga gerada por esses dois tipos de execuções, o de trabalho específico do benchmark e a externa ao *benchmark*, resultam na total carga de trabalho existente no sistema a ser analisado.

A carga de trabalho deve refletir ao máximo os procedimentos e recursos utilizados na aplicação na hora de sua utilização, o que vem a dar credibilidade e realidade aos resultados adquiridos para análise de seu desempenho. Para a análise de banco de dados, Costerano (1997), especifica alguns parâmetros que devem ter uma atenção maior na hora da descrição das operações que constituem a carga de trabalho, como: o tipo da transação seja ela uma seleção ou atualização, os predicados utilizados como números e tipos, os valores finais e intermediários para os resultados das transações e a frequência relativa das transações.

Sabendo da importância que a carga de trabalho tem no processo de análise do desempenho de um sistema, MENDES (2010) especifica uma metodologia a ser seguida no momento da definição da carga de trabalho, como:

- a) **definir a perspectiva da análise:** que objetos são necessários para a análise e quais devem ser levados em consideração, como SGBDs, SO, servidores WEB, entre outros;
- b) **identificar os componentes básicos:** que tipo de transação são necessária para a composição da carga de trabalho específica, transações de leitura, de escrita;

- c) **definir os parâmetros que melhor definam a carga de trabalho apresentada:** definição características como a intensidade , como a quantidade de registros em um SGBD e a demanda, tempo de utilização de partes do sistema, necessária para a carga de trabalho;
- d) **monitorar o sistema e coletar dados:** se possível e necessário, o monitoramento do sistema por um certo período ajuda a obter valores muito próximos do real a parâmetros já definidos;
- e) **particionar a carga:** conforme a grandiosidade e a complexidade do sistema, pode ser necessário dividir a carga de trabalho em classes que tenham características particulares semelhantes como o acesso a recursos, o modo de processamento, entre outros.

Com isso, uma base metodológica pode ser utilizada para que se chegue à descrição e caracterização da carga de trabalho de uma forma a cada vez se aproximar do real. Para a então construção do modelo da carga de trabalho algumas técnicas que podem ser utilizadas, como o uso de histogramas, especificação de médias de dispersão, etc.

4.3 OSDB e AS³AP

Criado pela Compaq Computer Corporation®, o *benchmark* OSDB tinha como propósito avaliar as taxas e I/O (entrada/saída) e o poder de processamento sobre a plataforma GNU Linux/Alpha. Um *benchmark* de código aberto, que teve sua implementação baseada no AS³AP, diferenciando em alguns aspectos como quantidades de módulos e quantidade de métricas retornadas. O AS³AP possibilita uma única métrica para análise dos seus resultados, que é o tamanho máximo do banco de dados suficiente

para completar o teste em menos de 12 horas, já o OSDB possibilita a análise através de outras características como o número de linhas retornadas por segundo e o tempo de resposta das consultas. (PIRES; NASCIMENTO; SALGADO, 2006)

O AS³AP é dividido em dois módulos, monousuários e multiusuário, já o OSDB, além desses dois módulos, incorporou outro chamado carga e estrutura, possuindo assim três módulos:

- a) **monousuário**: este módulo realiza os testes de desempenho das seleções, junções, projeções, agregações, visões, atualizações e exclusões, realiza teste também de integridade de chaves estrangeiras e inserções de chaves duplicadas. Segundo (PIRES; NASCIMENTO; SALGADO, 2006), o ordem de execução das consultas é estruturado de forma a não beneficiar a utilização de dados em *cache*;
- b) **multiusuário**: este módulo simula diferentes perfis de carga de trabalho, sendo que, em cada perfil, um número variado de processos são executados de forma concorrente, o que simula vários usuários conectados simultaneamente a uma base de dados. Para melhores resultados, o número de perfis utilizados nessa análise deve ser um quarto do tamanho da base de dados utilizada, ou um perfil para 4MB da base de dados;
- c) **carga e estrutura**: inclui a criação e carga de tabelas a partir de dados armazenados em arquivos textos e a criação de índices clusterizados e não-clusterizados.

4.4 BENCHMARKSQL

O *benchmarkSQL* é uma ferramenta benchmark utilizada na análise de desempenho de banco de dados implementada em linguagem de programação Java e disponibilizada com código aberto. Seu desenvolvimento teve base as especificações do TPC-C, desenvolvido pela *Transaction Processing Performance Council* (TPC), uma organização sem fins lucrativos que desde 1988 disponibiliza uma série de especificação para benchmark, como TPC-A, TPC-B, TPC-C, TPC-D, entre outros. Cada especificação se diferencia na forma e na maneira de em que as transações são executadas em cada teste.

Por se desenvolvida em Java, essa ferramenta faz a utilização de drivers *Java Database Connectivity* (JDBC) para realizar a comunicação entre o benchmark e o banco de dados, tornando-o compatível com uma vasta gama de SGBDS e podendo ser utilizado em sistemas operacionais como o Windows e o Linux (LIMA, 2008).

A utilização dessa ferramenta benchmark, segundo (CARNEIRO; MOREIRA; FREITAS) concentra-se em quatro etapas principais, como:

- a) **criação das tabelas:** a criação das tabelas do banco de dados que fazem parte do método TPC-C através da execução de um arquivo executável contido na ferramenta benchmark denominado runSQL.bat no Windows e runSQL.sh no Linux, onde, alguns parâmetros como o nome do arquivo de criação do SGBD e o comando que indicara a criação das tabelas;
- b) **carga das tabelas:** essa etapa é realizada a partir da execução do arquivo loadData.bat no Windows e load.Data.sh no Linux, gerando dados de forma aleatória nas tabelas criadas;

- c) **criação de índices:** nesta etapa são definidos os índices e as chaves que identificam cada tabela do banco de dados;
- d) **execução da ferramenta:** após as três etapas anteriores, a ferramenta benchmark pode ser executada, para tal, é necessário informar o nome do arquivo de configuração do banco de dados utilizado. Com isso, a ferramenta se mostra a partir de uma interface gráfica.

A interface gráfica dessa ferramenta é similar a outras janelas encontradas em ambiente Windows, onde alguns parâmetros devem ser configurados conforme o tipo de análise que se deseja executar, como a quantidade de máquinas utilizadas nos testes e o tipo de execução, que pode ser por número de transações ou por minuto, conforme a figura 4.

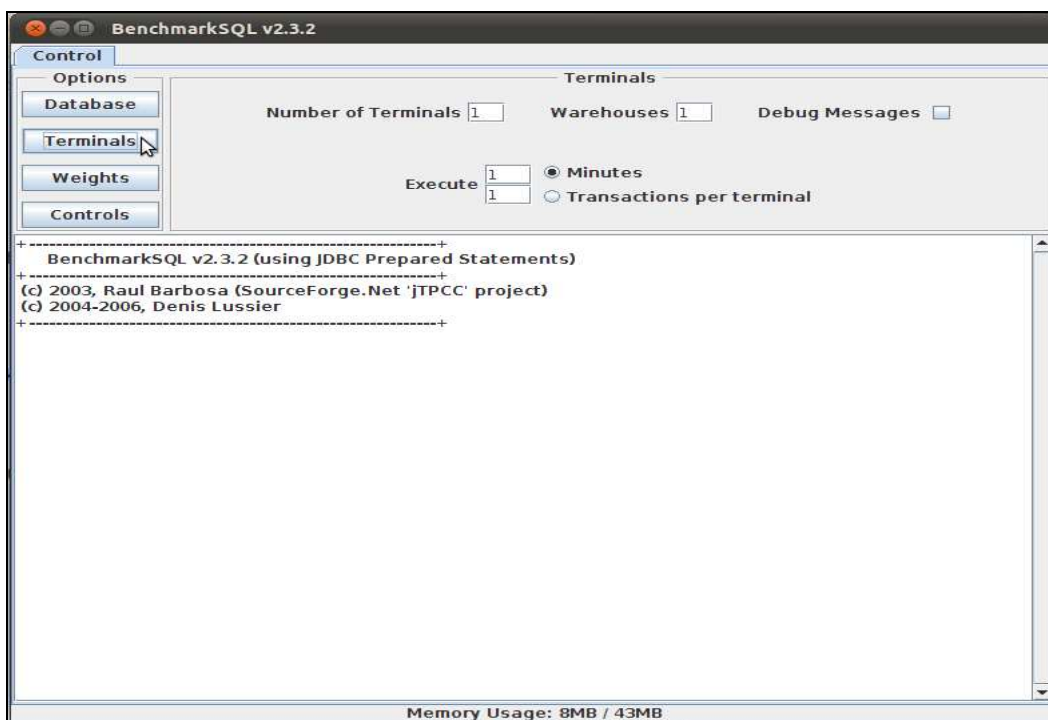


Figura 4. Interface gráfica do Benchmarksql.
Fonte: Dados do pesquisador.

Com essas e outras configuração definidas, pode dar início a execução da ferramenta a partir da criação dos terminais, opção *Create Terminals* e da execução das transações, *Start Transactions*.

A partir dos resultados obtidos através dessa ferramenta, se torna possível a realização do trabalho de análise de desempenho entre bancos de dados, onde neste trabalho se optou pela utilização e posterior comparação entre os SGBDs PostgreSQL e Informix, com base de dados de imagens em um sistema operacional Linux.

5 TRABALHOS CORRELATOS

Segundo o referencial bibliográfico realizado, pode se encontrar na atualidade diversos trabalhos que visam à avaliação de desempenho em diferentes produtos e nas mais diversas formas. Neste capítulo, serão mostrados alguns trabalhos que levam em consideração a avaliação de desempenho em SGBDs.

O trabalho realizado por BONFIOLI (2006) para a obtenção do título de bacharel em Ciência da Computação na Universidade Federal de Lavras, Minas Gerais, tem como objetivo avaliar o desempenho do PostgreSQL utilizando duas implementações diferentes ao um mesmo sistema, o relacional e o objeto relacional. Onde, foram avaliados desde métodos e facilidades de implementação quanto o próprio desempenho.

Já o trabalho de conclusão de curso desenvolvido por ALMEIDA (2008), no Centro Universitário Feevale, Novo Hamburgo, Rio Grande do Sul, que faz referencia a utilização de servidores virtuais onde permite uma melhor administração dos recursos disponíveis da área de TI. Com isso, realizou uma avaliação de desempenho nos SGBDs Oracle e SQL Server, analisando como reflete o seu comportamento em um ambiente virtual e em quais situações essa ambiente virtual se torna viável.

O trabalho realizado por LIMA (2008) para a obtenção do grau de mestre no programa de Pós-Graduação em Informática, desenvolvido na Universidade Federal do Paraná, tem como foco a execução distribuída de *benchmark* em SGBDs. No qual, segundo o autor, a execução centralizada em multiusuários em alguns casos pode influenciar nos resultados, vistos que os recursos computacionais da maquina utilizada podem se esgotar. Assim, ele apresenta algumas alternativas para a solução deste problema, como a execução de *benchmarks* distribuídos.

6 ANÁLISE COMPARATIVA DE DESEMPENHO ENTRE SGBDS POSTGRESQL E INFORMIX NO ACESSO A BASE DE DADOS DE IMAGENS

O trabalho aqui apresentado tem como principal objetivo realizar uma análise de desempenho entre os SGBDS PostgreSQL e Informix. Para alcançar tal objetivo, uma ferramenta *benchmark* contará com uma carga de trabalho composta por arquivos de imagens para realizar as avaliações. Por fim, será realizada uma comparação do desempenho entre ambos.

6.1 PROCEDIMENTOS METODOLÓGICOS

Para a definição deste trabalho, foi realizado um levantamento bibliográfico referentes a bancos de dados, que englobavam modelos mais utilizados, tipos de fontes de dados, entre outros. Com isso, sentiu-se a necessidade de realizar uma avaliação de desempenho entre SGBDs, visto que o mesmo interfere diretamente no funcionamento de uma aplicação.

O PostgreSQL versão 8.4.4 foi escolhido para a comparação pelo fato de ser um SGBD livre, ou seja, sem custos para sua utilização, e que emprega a utilização de modelos de dados objeto-relacional. Já o Informix Innovator se deu também pelo fato de possuir uma versão livre de banco de dados que também possibilita a utilização de modelos de dados do tipo objeto-relacional. Porém, o ponto forte de sua escolha se deu pelo fato de ser um produto da IBM que, com o levantamento do referencial bibliográfico, notou-se pouca utilização do mesmo em nível de pesquisas e desenvolvimentos científicos. O que nos faz entrar na dúvida de que a sua fraca

utilização esta relacionada ao seu desempenho, ou pela cultura de utilizar produtos mais conhecidos.

Para se obter valores passíveis a uma comparação confiável, seja ela qual for à área, uma série de requisitos devem ser levantados e definidos como padrão a fim de não influenciar no desempenho de algum dos lados. Com isso, os testes aqui realizados levaram em consideração alguns parâmetros como é descrito a seguir.

A metodologia de comparação se deu pelo modelo experimental ou empírico, que é a análise do próprio sistema com a utilização de uma ferramenta *benchmark*, o que torna a análise mais precisa e confiável. Assim, o banco de dados foi criado e os arquivos utilizados como carga de trabalho foram do tipo JPEG.

Os bancos de dados utilizados permitem o armazenamento de informações tanto estruturais, como *strings* e inteiros, quanto binários, *Binary Large Object* (BLOB). As imagens utilizadas nos testes foram todas do tipo JPEG, onde, por ser objeto do tipo BLOB, o formato não apresentou como grande ponto de significância dos resultados. Por outro lado, as imagens foram diferenciadas pelo seu tamanho, onde uma imagem tinha dimensões de 1024x1024 *pixels* com 582KB e outra de 1280x1024 *pixels* com 1MB de tamanho em disco.

O *benchmark* utilizado foi o BenchmarkSQL 2.3.2, por ser uma ferramenta compatível com os bancos de dados utilizados nesse trabalho e com o SO (sistema operacional). Desenvolvido em *Java*, essa ferramenta é apresentada ao usuário através de interfaces gráfica semelhante a outras encontradas em ambiente Windows, facilitando sua configuração e utilização. Interfaces que estão sempre em constante desenvolvimento e evolução para diminuir a complexidade de interação entre aplicações e usuários.

Todos os testes foram realizados utilizando uma mesma máquina, um computador *Notebook Toshiba Satellite* modelo L305-S5933, possuindo um processador Intel Pentium Dual Core com velocidade de 2.16 GHz, 1 MB em cachê, disco rígido SATA de 250 GB e 5400RPM (rotações por minuto) e 3 GB de memória RAM. O sistema operacional utilizado foi o GNU/Linux com distribuição UBUNTU 11.04, com o *kernel* versão 2.6.32 e com 1 GB de partição *swap*.

6.2 TESTES

Para a execução da ferramenta *Benchmarksql*, alguns parâmetros devem ser configurados conforme a execução desejada, assim a tabela 1 mostra as configurações utilizadas na ferramenta.

Tabela 1. Configuração do *Benchmarksql*

Payment %	43
Order-Status %	4
Delivery %	4
Stock-Level %	4
Number of Terminals	1
Minutes	5
Warehouses	1

Essa ferramenta possibilita tanto a execução por quantidade de transações por terminais, quanto por tempo de execução. Com isso, optou em realizar a execução em um tempo de 5 minutos cada, onde esse tempo deu como retorno uma quantidade média de 590.000 transações, sendo suficiente para a análise em questão.

Enriquecendo ainda mais a análise, será executada uma carga de trabalho sobre os bancos de dados com quantidades diferentes de imagens, em análises com uma imagem, oito, dezesseis, trinta e duas, e sessenta e quatro imagens, ambas com imagens de 585KB e 1MB.

Para que se obtenham resultados confiáveis e precisos, além do ambiente de testes, o número de replicações realizadas é essencial para que se obtenham valores passíveis a uma comparação confiável. Para essa descoberta, BARBETETTA; REIS; BORNIA, (2004) descrevem uma seqüência de cálculos para que se obtenha tal resposta.

Antes de dar início aos cálculos, uma série de testes aleatórios foram executados, 10 no PostgreSQL e 10 no Informix, onde foram armazenados os resultados finais para servir de base para os cálculos conforme mostra a tabela 2.

Tabela 2. Resultados de testes aleatórios.

Teste	Informix	PostgreSQL
1	4240,00	6012,00
2	4256,00	5945,00
3	4285,00	5932,00
4	4403,00	5943,00
5	4525,00	5968,00
6	4301,00	6232,00
7	4274,00	5974,00
8	4270,00	5921,00
9	4265,00	5813,00
10	4212,00	5899,00

O calculo principal é mostrado pela formula 1, onde (n_0) representa o número de amostras a ser realizada. Para realizar o calculo, algumas variáveis são

necessárias, como o valor de Z tabelado ($z_{\alpha/2}$), o valor da variância (σ^2) e valor da distancia máxima do valor populacional (E_0) (BARBETETTA; REIS; BORNIA, 2004).

$$n_0 = \frac{Z_y^2 \sigma^2}{E_0^2} \quad (1)$$

A descoberta dos valores necessários para a realização do calculo começa com os valores representados na tabela 2, onde calcula-se a média aritmética, sendo 4520,7 para o Informix e 6093,7 para o PostgreSQL, com isso, podese achar o valor da variância, conforme a formula 2.

$$n_0 = \frac{\sum_{i=1}^n d_i^2}{n - 1} \quad (2)$$

Para se calcular a variância, é subtraído o resultado de cada teste da média aritmética calculada anteriormente, elevando seu resultado ao quadrado.

O somatório desses valores é dividido por 9 , ($n-1$), n representa a quantidade de testes realizados. Assim, obtence o valor da variância (σ^2), onde no PostgreSQL foi de 11660,54 e no Informix de 8604,99.

Com esse valor em mãos, é possivel saber o valor do desvio padrão, que representa a variação dos dados por volta da média. O desvio padrão é resultado da raiz quadrada da variância, sendo assim, 107,98 para o PostgreSQL e 92,76 para o Informix sendo calculado conforme a formula 3.

$$\sigma = \sqrt{\sigma^2} \quad (3)$$

O próximo passo é calcular o valor do Z tabelado, sendo ele comum para ambos os bancos de dados. Para realizar esse cálculo, é necessário informar o valor da significância (α), definido pelo próprio desenvolvedor dos testes, onde para ele foi definido o valor de 0,05. Sendo esse o valor, é possível descobrir o nível de confiança conforme o cálculo 3.

$$y = 1 - \alpha = 1 - 0,05 = 0,95$$

Conforme o cálculo, o nível de confiança é de 95%, mostrado na figura 12.

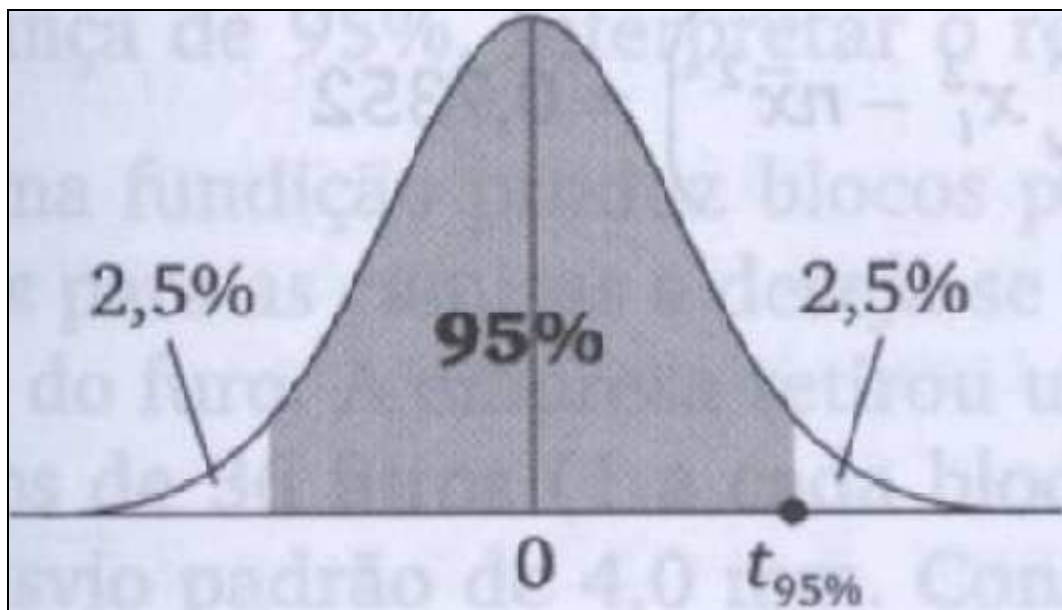


Figura 5. Nível de confiança de 95%
Fonte: BARBETTA; REIS; BORNIA (2004).

Com o valor da variância (α) que é de 0,05, é possível descobrir o valor do Z tabelado, consultando o anexo 1, linha z, coluna 0,025 da tabela.

Com isso, o valor do Z tabelado é de 1,96, encontrado na coluna 0,025 pelo fato de o resultado do valor da significância dividido por 2 determina a coluna a ser pesquisada. Outro valor necessário para o cálculo da fórmula principal é a o do erro estatístico (E), onde este é definido pelo desenvolvedor do cálculo representando a faixa de valor com maior chance de serem verdadeiros. O valor dado ao erro estatístico é $E = 25$.

Descoberto todas essas variáveis, é possível calcular o tamanho das amostras para o PostgreSQL e para o Informix, conforme a fórmula 1 dada anteriormente. Como resultado, o número mínimo de amostras para o Informix é de 53 e para o PostgreSQL é de 72.

Com base nos resultados, serão executados 200 testes, sendo 100 para o PostgreSQL e 100 para o Informix, 50 com imagens de 582K com resolução 1024x1024, e outros 50 com imagens de 1MB e resolução de 1280x1024, onde, a cada 10 testes realizados, a quantidade de imagens utilizadas aumenta gradativamente, passando de uma para oito, dezesseis, trinta e duas, e sessenta e quatro imagens respectivamente.

Dado início aos testes, foi avaliado o desempenho do Informix perante uma carga de trabalho com imagens de resolução de 1024x1024, tipo JPEG com 582 KB, onde, o resultado pode ser analisado na tabela 3.

Tabela 3. Resultados das análises com imagens de 582KB no Informix.

Informix					
Teste	1 Imagem	8 Imagens	16 Imagens	32 Imagens	64 Imagens
1	4186	4160	4126	4276	4374
2	4245	4610	4498	4647	4442
3	4608	3649	4508	4397	4403
4	4720	4529	4574	4340	4338
5	4720	4568	4528	3997	4273
6	4344	4105	4372	4473	3743
7	4945	4368	4534	4366	4248
8	4645	4105	4603	4472	4274
9	4510	4442	4607	4187	4255
10	4284	4537	4505	4247	4215
Média	4520,7	4307,3	4485,5	4340,2	4256,5

Como parte da análise total do Informix, a tabela 4 mostra os resultados das outras 50 avaliações que se deram com imagens com resolução de 1280x1024, tipo JPEG com tamanho de 1MB.

Tabela 4. Resultado das análises com imagens de 1MB no Informix.

Informix					
Teste	1 Imagem	8 Imagens	16 Imagens	32 Imagens	64 Imagens
1	4397	4509	4104	4438	4156
2	4682	4469	4475	4539	4276
3	4680	4786	4469	4573	4340
4	4241	4608	4251	4314	4250
5	4610	4641	4429	4306	4314
6	4749	4538	4412	4307	3676
7	4717	4193	4412	4279	3741
8	4569	4372	4607	4538	4507
9	4188	3964	3890	4369	3917
10	4610	4407	3743	4369	3917
Média	4544,3	4448,7	4279,2	4403,2	4109,4

Nota-se que na medida em que a quantidade de imagens aumenta, bem como o seu tamanho, o banco de dados sente e o desempenho sofre pequenas alterações. As 100 análises no Informix foram executadas e com base nos resultados, o desempenho médio do mesmo foi de 4369,5 linhas por segundo.

A análise do PostgreSQL teve início com a mesma carga de trabalho inserida nos testes iniciais do Informix, com imagens de 1024x1024 e 582KB, a tabela 5 mostra os resultados do PostgreSQL.

Tabela 5. Resultados das analisse com imagens de 582KB no PostgreSQL.

PostgreSQL					
Teste	1 Imagem	8 Imagens	16 Imagens	32 Imagens	64 Imagens
1	6172	5657	6815	5543	5815
2	5876	7054	5547	5589	4904
3	6178	7045	5645	5546	5082
4	6171	6510	5435	5549	5443
5	5819	5598	5647	6052	5444
6	6109	5705	5435	5445	5296
7	5934	6112	5820	5441	5402
8	6098	5649	5395	5493	5493
9	5695	5593	5548	5599	5444
10	6885	5550	5702	5879	5204
Média	6093,7	6047,3	5698,9	5613,6	5352,7

Conforme a métrica utilizada, após as 50 replicações, as imagens utilizadas como carga de trabalho são alteradas para imagens de 1280x1024, 1MB, assim, os resultados são mostrados na tabela 6.

Tabela 6. Resultados das análises com imagens de 1MB no PostgreSQL.

PostgreSQL					
Teste	1 Imagem	8 Imagens	16 Imagens	32 Imagens	64 Imagens
1	5713	5652	5491	5812	5815
2	6123	5649	5486	5764	5211
3	5967	5548	5162	5492	5298
4	5704	5594	5555	5399	5352
5	6054	5763	5708	5156	5252
6	6129	5602	5599	5544	5207
7	5876	5601	5444	5252	5255
8	5816	5708	5487	5451	5209
9	5986	5696	5496	5451	5350
10	6150	5645	5767	5648	5706
Média	5951,8	5645,8	5519,5	5496,9	5365,5

Com base em todas as avaliações executadas, o PostgreSQL teve um desempenho médio de 5678,4 linhas por segundo. Desempenho esse em um nível superior ao do Informix, que serão detalhados no tópico a seguir.

6.3 RESULTADOS

Com o término dos testes realizados através da ferramenta Benchmarksql nos bancos PostgreSQL e Informix, os resultados serão aqui apresentados. Como o número de amostras mínimas para a análise em questão foi significativamente alto, com o decorrer dos testes o número de imagens bem como o seu tamanho foi sendo elevado gradativamente, até que se completassem 100 replicações para cada banco, como relatado anteriormente.

Os primeiros resultados apresentados são referentes a 10 replicações com uma imagem de 582KB e mais 10 com uma imagem de 1MB, totalizando 20 replicações,

ambas no formato JPEG. Como resultado, o Informix apresentou-se uma diferença de 1490,25 linhas por segundo a menos em relação ao resultado do PostgreSQL, obtendo um desempenho 24,74% inferior conforme mostra a figura 6.

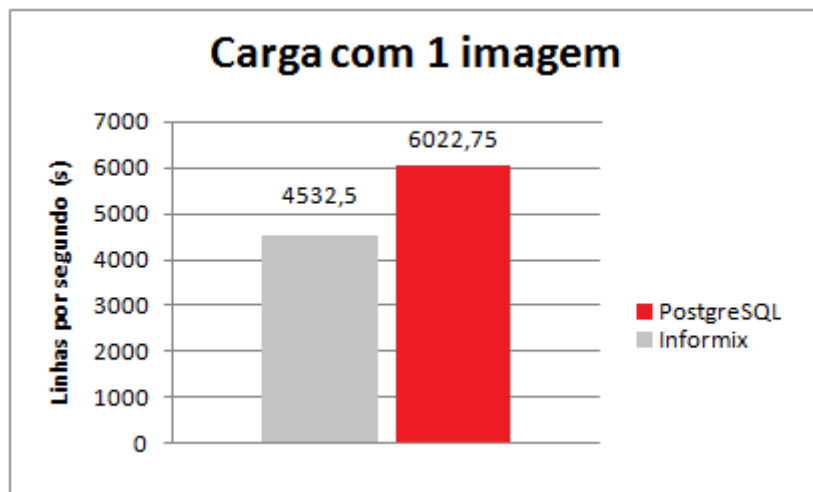


Figura 6. Gráfico comparativo do desempenho com uma imagem.
Fonte: Dados do pesquisador.

Em seguida a quantidade de imagens na carga de trabalho foi elevada, totalizando oito imagens iguais as das análises anteriores, assim outras 20 análises foram realizadas. O Informix manteve um desempenho inferior ao do PostgreSQL, como mostra a figura 7, uma diferença de 1468,55 linhas por segundo, 25,12%.

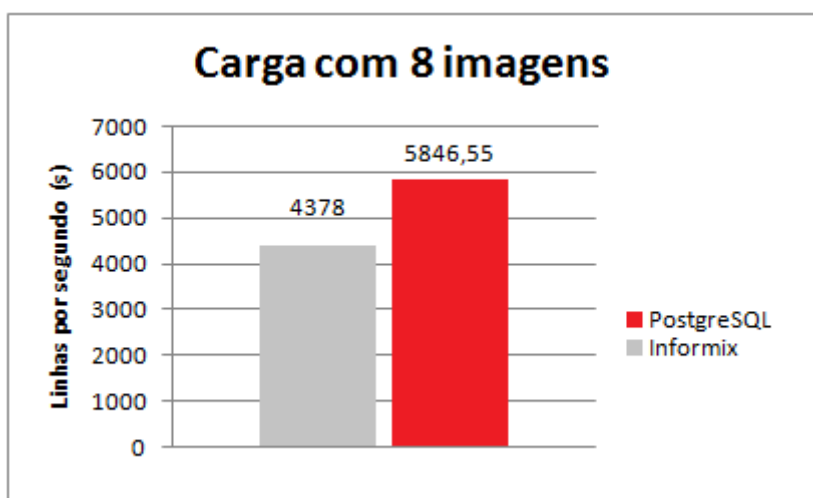


Figura 7. Gráfico comparativo de desempenho com carga de 8 imagens.
Fonte: Dados do pesquisador.

Com dezesseis imagens postas como carga de trabalho, o Informix apresentou um desempenho muito semelhante aos resultados obtidos nas 20 análises anteriores. Já o PostgreSQL manteve uma queda de rendimento, como mostram as figura 7 e 8.

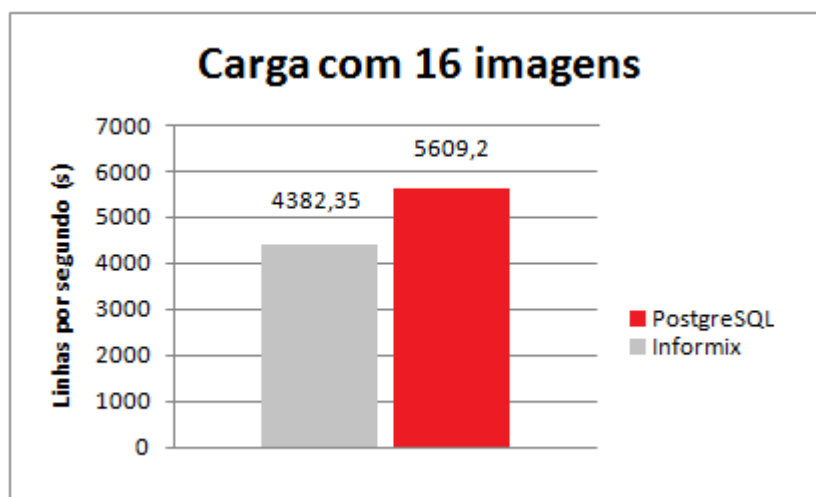


Figura 8. Gráfico comparativo de desempenho com carga de 16 imagens.
Fonte: Dados do pesquisador.

Até esse ponto da análise, onde se totalizou 60 replicações já executadas, mais de 50% do trabalho, o Informix obteve um desempenho médio de 4430,05 linhas por segundo contra 5826,16 linhas do PostgreSQL.

Dando continuidade a análise, novamente a carga de trabalho foi elevada, agora para trinta e duas imagens, onde novamente o Informix demonstrou uma pequena queda de rendimento se comparado com a queda de rendimento do PostgreSQL como mostram as figuras 8 e 9.

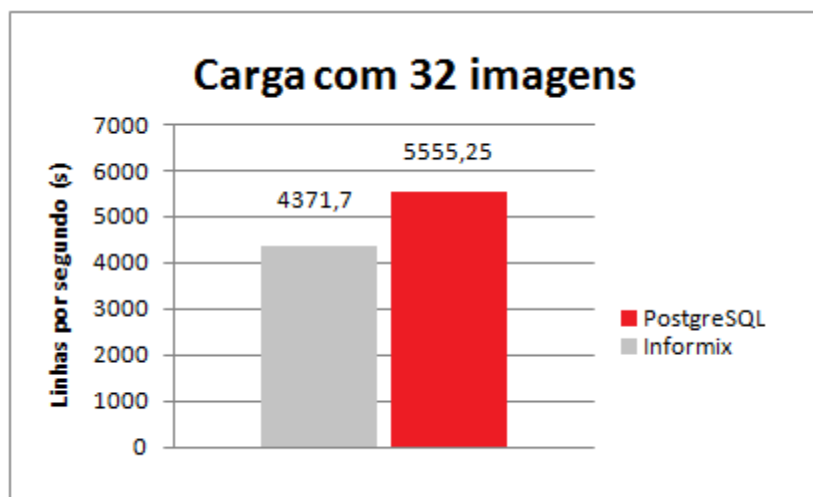


Figura 9. Gráfico comparativo de desempenho com carga de 32 imagens.
Fonte: Dados do pesquisador.

Neste caso específico, a diferença de rendimento do Informix em relação ao PostgreSQL foi de 23,30%.

Finalizando o processo de análise de desempenho e totalizando as 100 replicações, os bancos de dados foram submetidos a uma carga de trabalho com sessenta e quatro imagens. Com isso os resultados demonstrados na figura 10 mostram que o Informix obteve um desempenho inferior, com exatas 1176,15 linhas processadas por segundo a menos que o PostgreSQL, uma diferença de 21,94%.

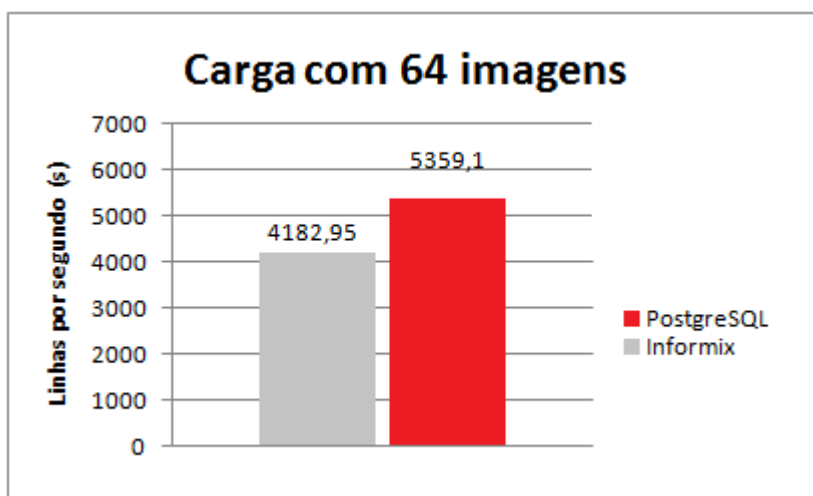


Figura 10. Gráfico comparativo de desempenho com carga de 64 imagens.
Fonte: Dados do pesquisador.

Por fim, pode-se notar que o PostgreSQL obteve um desempenho superior ao Informix. Porém, o mesmo foi o que apresentou maior queda de rendimento na medida em que a quantidade de imagens foi gradativamente aumentando, apresentando uma queda de 11,01% se comparada com as primeiras 20 replicações. Com o Informix, na mesma situação, a queda foi de apenas 7,71%.

Essa queda de desempenho continua a se repetir se separarmos as replicações realizadas com imagens de 582KB e 1MB, onde foram 50 para cada, como mostram as figuras 11 e 12.

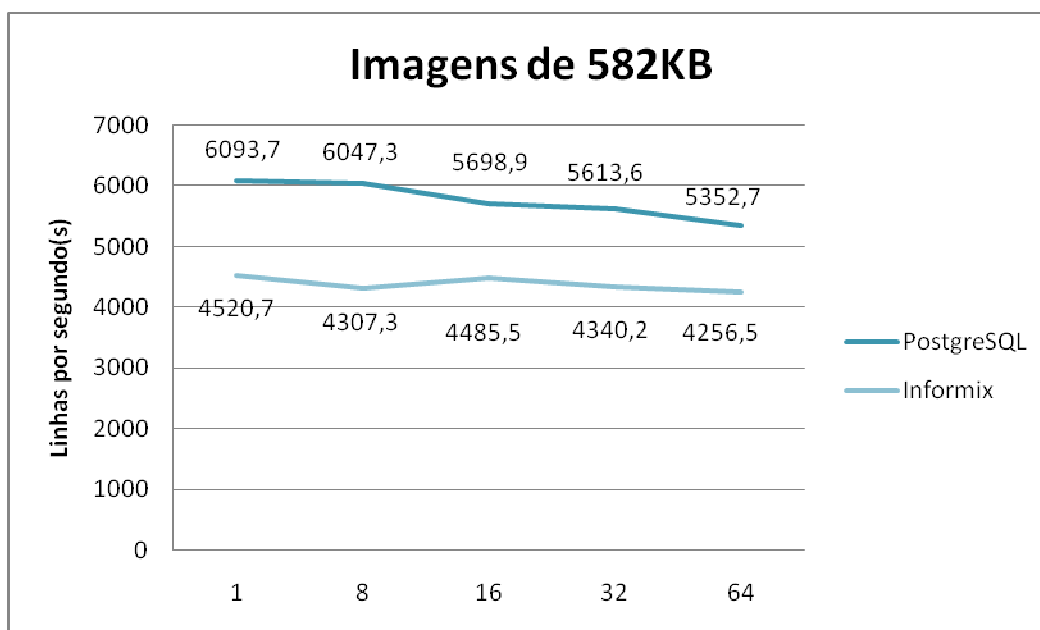


Figura 11. Comparação do desempenho em cargas de imagens com tamanho de 582KB.
Fonte: Dados do pesquisador.

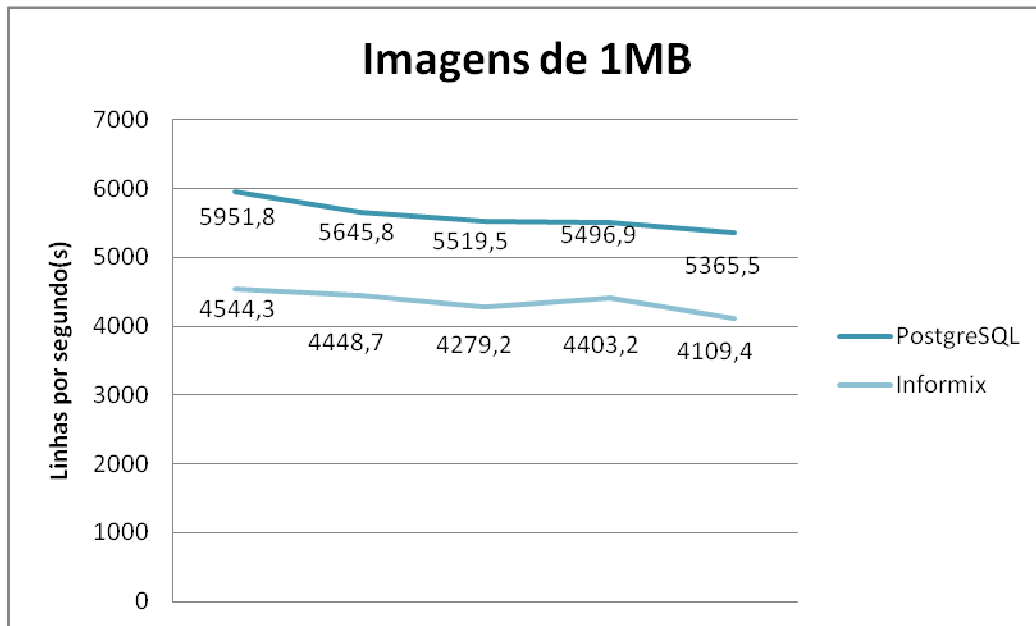


Figura 12. Comparação do desempenho em cargas de imagens com tamanho de 1MB.
Fonte: Dados do pesquisador.

O desempenho entre as duas aplicações ao final das análises pode ser visto e comparado como mostra a figura 13, onde o PostgreSQL manteve o desempenho superior ao do Informix.

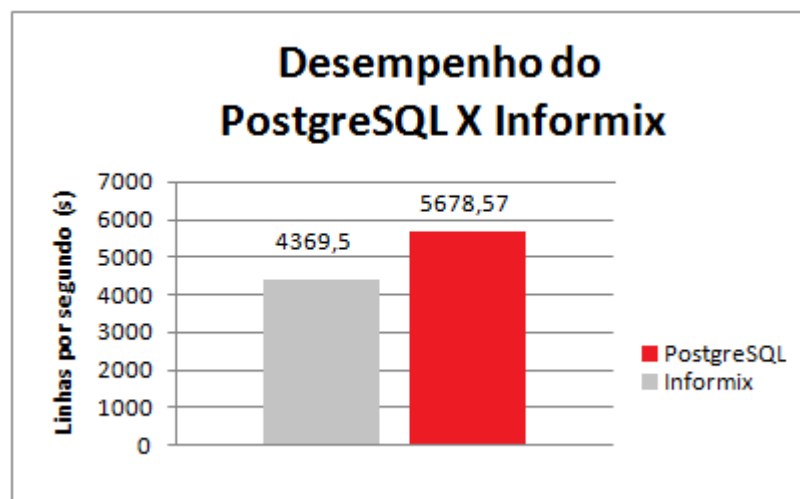


Figura 13. Comparação do desempenho do Informix X PostgreSQL.
Fonte: Dados do pesquisador.

Ao final, nota-se que a quantidade de amostras é fundamental para uma análise justa e confiável, onde se pode verificar que o resultado ao final de 100 replicações é consideravelmente diferente das primeiras 20 ou 40 replicações iniciais. Assim, o PostgreSQL obteve um desempenho 5678,57 linhas por segundo, contra 4369,5 do Informix, uma diferença final de 23,05%.

CONCLUSÃO

Através do presente estudo pode-se discorrer sobre conceitos e características dos SGBDs PostgreSQL e Informix, e de algumas ferramentas *benchmarks* tendo como principal foco o Benchmarksql. Onde essas aplicações foram utilizadas no processo de avaliação de desempenho alvo estudo desde trabalho. A carga de trabalho submetida aos bancos foi composta por imagens, que no decorrer dos testes, o número de imagens e o seu tamanho foram elevados gradativamente, enriquecendo ainda mais o resultado final.

Com o decorrer do desenvolvimento desta, o objetivo específico foi alcançado, que era a comparação de desempenho entre SGBDOR no acesso à base de dados de imagens por meio da utilização de uma ferramenta *benchmark*. Onde pode ser comprovado no capítulo que se refere à realização dos testes.

Os objetivos específicos também foram sendo alcançados na medida em que o presente trabalho foi sendo desenvolvido, onde foram documentados conceitos referentes a bancos de dados e aos SGBDs utilizados, bem como a definição de possíveis ferramentas *benchmark* para a realização de avaliação de desempenho. A realização da análise foi iniciada logo após a definição dos estudos de casos e das métricas para a obtenção de resultados confiáveis e precisos. Com isso, se pode realizar uma comparação de desempenho entre os SGBDs envolvidos.

O desempenho final foi resultado de inúmeras amostras realizadas sucessivamente, onde o *benchmark* retornava como resultado da análise o número de linhas processadas em um período de tempo. Em todos os casos analisados, o PostgreSQL obteve um desempenho superior ao do Informix, tendo um desempenho final 23,05% maior que o concorrente.

Apesar disso, o Informix foi o SGBD que apresentou uma menor queda de rendimento na medida em que as amostras eram realizadas e o tamanho da carga era elevado. Com isso, esse trabalho pode servir como base para que desenvolvedores tomem a decisão de qual SGBD utilizar no desenvolvimento de uma aplicação, levando em consideração o tipo de informação manipulada.

Analisando todos os resultados e suas particularidades, nota-se que alguns trabalhos futuros podem ser desenvolvidos e servir para continuar a enriquecer o resultado da idéia inicial. Partindo do ponto que o Informix apresentou menor queda de rendimento à medida que a carga aumentava, trabalhos podem seguir a aumentar ainda mais essa carga até com arquivos diferentes e assim continuar a analisar esse desempenho.

Futuros trabalhos também podem ser desenvolvidos utilizando diferentes tipos e formatos de arquivos multimídias, como áudio ou vídeo, de sistemas operacionais como o Windows ou GNU/Linux. Também pode ser avaliado o desempenho entre bancos de dados distintos, com o auxílio de diferentes ferramentas *benchmark*. Trabalhos esses que caminham todos com um bem comum, a avaliação e comparação de desempenho entre SGBDs.

REFERÊNCIAS

- ALMEIDA, Gustavo Porto de. **Análise da utilização de SGBDs em servidores virtualizados no ambiente corporativo**. 2008. 73f. TCC (Sistemas da Informação) – Centro Universitário Feevale , 2008
- BARBETTA, Pedro Alberto; Reis, Marcelo Menezes; Bornia, Antonio Cezar. **Estatística**: para cursos de engenharia e informática. São Paulo: Atlas, 2004.
- BITTENCOURT, Danilo de Souza. **Banco de dados de imagens**: uma revisão da literatura. 2007. 57f. TCC (Ciência da Computação) – Universidade Federal de Lavras, 2007
- BONFIOLI, Guilherme Ferreira. **Banco de dados Relacional e Objeto-Relacional**: uma comparação utilizando Postgresql. 2006. 50f. TCC (Ciência da Computação) – Universidade Federal de Lavras, 2006
- CASANOVA, Marco Antonio. **Banco de dados geográficos**. Curitiba, PR: Mundo Geo, 2005.
- CARVALHO, Adriano Arlei de. **Fundamentação teórica para Processamento Digital de Imagens**. 2003. 18 f. TCC (Ciência da Computação) – Universidade Federal de Lavras, Lavras, 2003
- CIFERRI, Ricardo Rodrigues. **Um benchmark voltado à análise de desempenho de sistemas de informações geográficas**. Dissertação Mestrado (Ciência da Computação) – Universidade Estadual de Campinas, 1995.
- COSTERANO, Walter Paulo. **Caracterização de sistemas de banco de dados espaciais para análise de desempenho**. Dissertação Mestrado (Ciência da Computação) – Universidade Estadual de Campinas, Campinas, 1997.
- DATE, C. J. **Introdução a sistemas de bancos de dados**. 7. ed. Rio de Janeiro: Ed. Campus, 2000.
- CARNEIRO, Alessandro P.; MOREIRA, Julinao L.; FREITAS, Andre L. C.. **Tuning – Técnicas de Otimização de Banco de Dados**: Um Estudo Comparativo. Artigo. Universidade Federal do Rio Grande (UFRG), 2006.
- DELFINO, Sérgio R.; NUNES, Fátima L. S.; SPOTO, Edmundo S.. **Avaliação de desempenho de SGBDs gratuitos para construção de bases de imagens médicas**. In: IV SIMPÓSIO BRASILEIRO DE QUALIDADE DE SOFTWARE, Porto Alegre, 2005.
- DOCUMENTACAO do PostgreSQL. **Documentação do PostgreSQL 8.0.0**. The PostgreSQL Global Development Group, 2007. Disponível em: <www.sourceforge.net/projects/pgdocptbr>. Acesso em: set 2011.

DOE, Carlton. **Compare as edições do Informix Dynamic Server, Versão 11**. Saiba as diferenças em recursos, funções e modelos de precificação, 2011. Disponível em: <<http://www.ibm.com/developerworks/br/data/library/techarticle/dm-0801doe/#table>>. Acesso em: set 2011.

DURÃES, Dayane Helen Santos. **Análise de desempenho entre banco de dados Postgresql e Firebird na plataforma Windows 2000**. 2007. 42 f. TCC (Sistemas de Informação) – Universidade Estadual de Monte Carlos, Monte Carlos, 2007.

ELMASRI, Ramez; NAVATHE, Sham. **Sistemas de banco de dados: fundamentos e aplicações**. Rio de Janeiro: LTC, 2002.

FILHO, Ogê Marques. NETO, Hugo Vieira. **Processamento Digital de Imagens**. Rio de Janeiro: Brasport, 1999.

GONZALEZ, Rafael C.; WOODS, Richard E. **Processamento de Imagens Digitais**. São Paulo: Edgar Blucher Ltda, 2000.

GONZÁLEZ, Sahudy Montenegro. **Modelo para o Gerenciamento de Banco de Dados de Imagens**. Tese Doutorado (Engenharia Elétrica) – Universidade Estadual de Campinas, 2003.

HARRINGTON, Jan L. **Projetos de bancos de dados relacionais**. Rio de Janeiro: Campus, 2002.

KROENKE, David M. **Banco de dados: fundamentos, projeto e implementação**. 6 ed. Rio de Janeiro: LTC, 1999.

MACHADO, Felipe Nery Rodrigues; ABREU, Maurício Pereira de. **Projeto de banco de dados: uma visão prática**. 8.ed São Paulo: Érica, 2002.

MENDES, Vitor Mendes. **Utilizando benchmark para avaliação de desempenho em sistemas gerenciadores de banco de dados para base de dados de imagens**. 2010. 85 f. TCC (Ciência da Computação) – Universidade do Extremo Sul Catarinense, Criciúma, 2010

LIMA, Murilo Rodrigues de. **Execução distribuída de benchmarks em sistemas de banco de dados relacional**. Dissertação Mestrado (Ciência Exatas) – Universidade Federal do Paraná, Curitiba, 2008.

OLIVEIRA, Ronaldo Aparecido de; SILVA, João Fernando Custódio da. **Banco de imagens georreferenciadas no mapeamento terrestre móvel**. Bol. Ciênc. Geod., Curitiba, v. 7, n. 2, 2001.

OSDB. **The Open Source Database Benchmark**. Disponível em: <www.osdb.sourceforge.net>. Acesso em: 1 out 2011.

PEREIRA, Rafles Frausino. **Análise de desempenho de banco de dados utilizando benchmarks especializados**. 138 f. Dissertação de Mestrado (Ciência da Computação) – Universidade Estadual de Campinas, Campinas, 1990.

PIRES, Carlos Eduardo. NASCIMENTO, Rilson Oscar. SALGADO, Ana Carolina. **Comparativo de Desempenho entre Banco de Dados de Código Aberto**. Artigo. Universidade Federal de Pernambuco (UFPE), 2006.

POSTGRESQL. **PostgreSQL**. Disponível em: <www.postgresql.org>. Acesso em: 12 set. 2011.

RODRIGUES, Fernando. **Análise de desempenho e carga em acesso simultâneo a um banco de dados**. 2007. 64 f. TCC (Ciência da Computação) – Universidade do Extremo Sul Catarinense, Criciúma, 2007.

RODRIGUES, Ricardo. **Análise comparativa dos principais benchmarks voltados à análise de desempenho de SGBDOOs**. Unimar, Maringá, p.997-1016, 26 nov. 1997.

SCURI, Antonio Escaño. **Fundamentos da Imagem Digital**. Rio de Janeiro: Tecgraf/PUC, 1999.

SILBERSCHATZ, Abraham; KORTH, Henry F.; SUDARSHAN, S. **Sistema de banco de dados**. 3. ed. São Paulo: Makron Books, 1999.

SILVA, Fagundes Pereira de. **Um benchmark para sistemas de gerenciamento de workflow**. 91 f. Tese Mestrado (Engenharia Elétrica) - Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2004.

SILVA, Ricardo Czelusniak da. **Benchmark em banco de dados multimídia: análise de desempenho em recuperação de objetos multimídia**. 2006. 64 f. Dissertação Mestrado (Informática) – Universidade Federal do Paraná, Curitiba, 2006.

SUPRIANO, Anderson. **Avaliação de um Sistema de Gerência de Banco de Dados em Memória Principal para o uso em aplicações WEB**. Tese Mestrado (Engenharia da Computação) – Universidade Estadual de Campinas, Campinas, 2006.

VIEIRA, Marco; DURÃES, João; MADEIRA, Henrique. **Especificação e Validação de Benchmarks de Confiabilidade para Sistemas Transacionais**. IEEE Latin America Transactions, [s.i.], v. 3, n. 1, maio 2005.

WEISS, Emerson. **Desempenho de banco de dados para pequenas arquiteturas**. 51 f. Monografia Especialização (Engenharia de Software e Banco de Dados) - Universidade Estadual de Londrina, Londrina, 2008.

APÊNDICES

Apêndice 1. Artigo.

Avaliação de Desempenho no Acesso a Base de Dados de Imagens entre SGBDs PostgreSQL e Informix.

Samuel Luiz Goudinho¹, Paulo João Martins²

¹Acadêmico do Curso de Ciência da Computação, ²Professor do Curso de Ciência da Computação – Unidade Acadêmica de Ciências, Engenharias e Tecnologias – universidade do Extremo Sul Catarinense (UNESC) – Criciúma, SC – Brasil

samuel Luiz@hotmail.com.br, pjm@unesc.net

Abstract. *The fast and accurate information is critical to the sustainability of companies and organizations in such a competitive world. Thus, the present work, realized a performance comparison between the Management Databases Systems PostgreSQL and Informix. For the tests, were used image files, as workload, and the tool Benchmarksql, on GNU/Linux to measure performance. Obtaining reliable results, before starting to perform the analysis, it was necessary to make some random tests to calculate the minimum number of samples. During the analysis, every 10 tests the workload suffered a gradual increase. As a result, although the Informix suffer the lowest drop in performance according it was increased the size of the load, PostgreSQL obtained a superior final performance than that.*

Resumo. *A informação rápida e precisa é fundamental para a sustentabilidade de empresas e organizações em um mundo tão competitivo. Com isso, o presente trabalho, realizou uma comparação de desempenho entre os Sistemas Gerenciadores de Bancos de Dados PostgreSQL e Informix. Para a realização dos testes, foram utilizados arquivos de imagens, como carga de trabalho, e a ferramenta Benchmarksql, em ambiente GNU/Linux para medir o desempenho. Para obter resultados confiáveis, antes de começar a executar as análises, foi necessário realizar alguns testes aleatórios para calcular o número mínimo de amostras. No decorrer das análises, a cada 10 testes realizados a carga de trabalho sofria um acréscimo gradativo. Como resultado, apesar de o Informix sofrer a menor queda de rendimento na medida em que se aumentava o tamanho da carga, o PostgreSQL obteve um desempenho final superior aquele.*

1. Introdução

Um banco de dados esta relacionado com um conjunto de informações que relacionadas representam aspectos do mundo real. Já o SGBD é o conjunto de sistemas que possibilita tanto a criação quanto a manipulação de forma consistente dos dados. Com o passar do tempo, algumas aplicações estão perdendo o rótulo de aplicações

tradicionais e incorporando características que os classificam como aplicações multimídias. Isso porque as informações antes manipuladas em arquivos de textos não estão sendo mais suficientes para suprir as novas necessidades. Passando assim muitas vezes para arquivos de imagens, uma realidade presente em aplicação de informações médicas, informações geográficas, entre outras.

A partir dessas informações, essa pesquisa tem como objetivo a realização de uma avaliação de desempenho entre os SGBD PostgreSQL e Informix tendo imagens como fonte de dados. Uma ferramenta benchmark será utilizada para a realização dos testes.

1.1 Objetivo Geral

Comparar o desempenho de SGBDOR no acesso à base de dados de imagens por meio de uma ferramenta *benchmark*.

1.2 Objetivo Específico

Os objetivos específicos relacionados com essa pesquisa são:

- e) descrever e aplicar os conceitos de banco de dados orientado a objetos e objeto relacional;
- f) documentar modelos existentes e fazer uso de uma ferramenta *benchmark* para avaliação de SGBD;
- g) definir os estudos de casos a serem aplicados a análise de desempenho;
- h) estudar e aplicar as métricas de comparação entre SGBDs.

1.3 Justificativa

Um SGBD implica diretamente no funcionamento de uma aplicação, visto que todas as manipulações dos dados são realizadas conforme o próprio determina. Ele é responsável por disponibilizar um ambiente tanto eficiente quanto conveniente para a manipulação desses dados. DATE (2002) afirma que o SGBD não é o único componente de uma aplicação, porém é de longe o mais importante.

Com o advento de aplicações que utilizam imagens como fonte de dados, surge a necessidade de se determinar quais produtos se mostram eficiente quando submetidos a manipulação desses tipos de arquivos. Vale salientar que esses arquivos não eram considerados quando os SGBDAs foram projetados.

Com isso, será realizado uma avaliação de desempenhos entre os SGBDs PostgreSQL e Informix com imagens como fonte de dados, onde os resultados aqui mostrados podem ser pontos para futuras escolhas.

2. Banco de Dados e Sistema Gerenciador de Banco de Dados

Um banco de dados esta relacionado com uma coleção de dados persistente que relacionados entre si representam aspectos do mundo real, sendo interessante a um determinado grupo específico. Um Sistema Gerenciador de Banco de Dados é um conjunto de programas que possibilita tanto a criação quanto a manutenção de um banco de dados. Eles surgiram com o intuito de substituir e supri as necessidades e desvantagens antes encontradas com em programas de processamento de arquivos. Visando manter a integridade dos dados, é ele quem impõe restrições e formas de manipulação ao banco. Desde a sua criação, alguns modelos podem ser facilmente encontrados no mercado, onde podemos citar o modelo relacional, o orientado a objetos e o objeto relacional.

O SGBD Relacional trabalha com o armazenamento dos dados em tabelas, cada qual com uma identificação única. Os valores de uma tabela se relacionam através de ligações com valor de outras tabelas, fazendo assim o relacionamento entre os dados

e transformando dados em informação. Esse modelo é de longe o mais utilizado, porém, apresentando algumas dificuldades quando submetido à manipulação de dados complexos, como áudio, vídeo, imagens, etc.

O desenvolvimento do Sistema Gerenciamento Orientado a Objetos tomou como fundamento os conceitos dos bancos de dados tradicionais em junção com os recursos dos conceitos de programação orientada a objetos. Neste modelo, cada informação é armazenada na forma de objetos. Cada objeto é composto pelas variáveis que são os dados a serem armazenados e uma série de métodos que operam sobre esse objeto. Assim, possibilita a utilização de novos modelos de dados, considerados complexos ou não convencionais como números e textos. Esse modelo se baseia em algumas características que ajudam a entender melhor o seu funcionamento, como a persistência transparente, ausência de impedância mismatch e identidade do objeto.

A utilização desse modelo, para, Elmasri e Natvthe (2002), se dá por uma série de fatores como a crescente utilização de linguagens de programação orientada a objetos, uma realidade em plena ascensão. Bem como pelo fato de possibilitarem que desenvolvedores definam as estruturas dos objetos especificando quais operações podem ser aplicadas a esses objetos. Porém esses modelos de dados ainda não apresentam uma maturidade suficiente para ser incorporado a grandes volumes de dados (SILBERSCHATZ; KORTH; SUDARSHAN, 1999).

O modelo objeto relacional veio a unir o útil com agradável. Com a junção das características do modelo relacional com a orientação a objetos, esse modelo vem como uma evolução do modelo relacional. Para tal evolução, esse modelo se apóia em características como versatilidade, robustez e outras tecnologias do modelo relacional, com a flexibilidade e potencialidade tecnologia orientada a objeto (BITTENCOURT, 2007). Em ambos os casos, tanto no banco de dados orientado a objeto quanto no banco de dados objeto-relacional, o paradigma de orientação a objeto se faz presente em sua implementação, se diferenciando apenas na sua manipulação.

3 Base de Dados para Imagens

Com a crescente evolução dos SGBDs e das necessidades, os dados em um banco de dados, em muitas áreas, vem deixando de lado o caráter textual e incorporando arquivos como áudio, vídeo e imagens. Dentre eles os arquivos de imagens estão largamente os mais utilizados. Essas bases de dados são cada vez mais comuns em aplicação como SIG, em áreas da medicina, biologia, cartografia, entre outros.

Esses bancos de dados de imagens devem ter o poder tanto de trabalhar com grandes volumes de informações como também lidar com metadados, informações vinculadas a outras informações contidas no mundo digital. Podendo ser manipulados, esses metadados fornecem tanto facilidades para a recuperação de informações digitais, como também nos dá o poder de entender, lidar e informar opiniões sobre elas (BITTENCOURT, 2007).

Para Gonzales (2003), algumas soluções tentam dar um suporte maior para os SGBDs lidarem com imagens, como por exemplo:

- d) **integração virtual**: uma solução simples, que não faz utilização de armazenamento do arquivo no banco. Isso porque, permanecem devidamente armazenados no banco apenas informações textuais que fazem referência ao arquivo, como nome, tamanho, comentários, manipulados fora da aplicação.
- e) **pseudo ou semi-integração**: nesta solução o armazenamento do arquivo multimídia no banco de dados se faz necessário, porém, a

interpretação todas as manipulações sobre elas são feitas por meio de identificadores e é de responsabilidade da aplicação.

- f) **integração real:** aqui os arquivos são armazenados de forma agregada. O usuário tem todo o poder de manipular as informações usando funções do próprio SGBD, sem necessidade dos identificadores ou informação.

3.1 Imagens Digitais

Uma imagem pode ser representada de diversas formas no mundo real, que vão desde fotos com altas resoluções, pinturas de quadros até riscos em parede de cavernas. Para que essas imagens possam ser inseridas no mundo digital, elas são transformadas e constituídas de pontos distribuídos em uma matriz de linhas e colunas (x, y) , conforme a figura 2. Cada ponto ou *pixel* se refere a uma cor específica definida por meio de um coeficiente $f(x, y)$ (CARVALHO, 2003). Assim, pode se dizer que uma imagem é o resultado de uma matriz de pixels, onde cada índice de linhas e colunas determina um ponto da imagem, e o valor do elemento dessa matriz determina a cor, ou nível de cinza naquele ponto.

Dependendo da aplicação ou da finalidade, diferentes formatos de imagens podem ser adotados, como *Joint Photographic Expert* (JPEG) e *Microsoft Windows Bitmap* (BMP), os mais conhecidos e utilizados, porém, não os únicos, existem ainda outros formatos como o GIF, TIFF, entre outros.

4 Avaliação de Desempenho em SGBDs

Em qualquer aquisição, seja de um produto ou serviço, algumas características podem ser levadas em consideração para uma boa escolha, como preço, qualidade, desempenho, entre outros. Essa avaliação pode ocorrer dentro de um supermercado, na escolha de uma universidade, na compra de um carro, entre outros.

No ramo da ciência não é diferente, tanto produtos de hardware, softwares possuem características passíveis de comparação, encontradas a partir do que chamamos de análise de desempenho. Uma técnica que consiste em medir a eficiência sobre algumas determinadas funcionalidades, sobre algum determinado aspecto específico, considerando algum tipo de medida como tempo e distância (COSTERANO, 1997). No ramo da computação (SVOBODOVA, 1976 apud DURÃES, 2007) destaca que a técnica de avaliação de desempenho tem como objetivo verificar a efetividade com que os recursos de hardware são utilizados para atender aos objetivos do software. Onde essa avaliação pode ser comparativa, quando se compara desempenhos de sistemas diferentes, ou analítica, quando a comparação é realizada mediante parâmetros e configurações do próprio sistema.

Se tratando de avaliação de desempenho em SGBD, inúmeros fatores podem vir a interferir no seu desempenho, como o sistema computacional utilizado, hardwares e softwares, como o próprio desenvolvimento tanto do banco de dados e do sistema que o utilizará. Além disso, Costerano (1997), afirma que existem fatores que interferem no desempenho do SGBD e que os usuários não podem fazer nada, como o próprio código que o mesmo foi implementado. Porém, destaca também que algumas modificações de parâmetros, criação de índices e agrupamentos podem ser realizados com o intuito de otimizar a utilização dos recursos.

Na atividade da análise de desempenho, vários autores como Ciferri (1995), Costerano (1997) e Pereira (1990), destacam a utilização de três modelos que dever servir como fundamento:

- a) **modelo analítico**: utilização de um conjunto de função matemáticas que aliadas a algoritmos que os resolvem e relacionam medidas de desempenho de um SGBD. Para Pereira (1990), este modelo tende a resultados menos precisos em virtude da utilização de registros de bancos e cargas de trabalho hipotéticas.
- d) **modelo de simulação**: procura reproduzir as atividades de um SGBD a partir de um conjunto de hipóteses mais prováveis. A simulação é realizada com base a entrada e saída (I/O) de dados. Porém tende a resultados menos precisos, visto a dificuldade de se definir os parâmetros de um sistema (COSTERANO, 1997);
- e) **modelo experimental**: utiliza o próprio SGBD para a obtenção dos resultados, o que aumenta a precisão dos resultados. Este modelo se baseia em duas técnicas, a de monitoramento e a de *benchmark*. Para Costerano (1997), esta técnica pode ser utilizada para comparação de diferentes sistemas por ser padronizados, contendo testes invariáveis e bem definidos.

Em alguns casos específicos, corre a utilização de modelos híbridos, a utilização de características e vantagens de dois ou mais modelos com a finalidade de uma representação mais precisa e realista do sistema. Para Ciferri (1995) e para Costerano (1997) essa análise de desempenho pode tornar real algo passível de comparação, podendo responder questões como uma análise de custo benefício, como a comparação entre tecnologias, entre outros.

4.1 A técnica de Benchmark

Para a realização de avaliação de desempenho, tanto em SGBD quanto a outros dispositivos computacionais, é utilizada uma técnica de *benchmark*. Uma ferramenta especialista em testes de desempenho, onde em SGBDs, possibilita a definição de uma série de transações, que quando executadas geram uma medida capaz de servir como base para comparações. Estas transações são chamadas de carga de trabalho, e a sua utilização é fundamental para uma análise justa e coerente. O fato de esta técnica ser executada em junção com próprio sistema a ser analisada dá o crédito necessário para uma análise muito próxima do real.

Existem diversas implementações de ferramentas *benchmark* disponíveis no mercado, como o AS³AP, OSDB, TPC, Wisconsin, Benchmarksql, entre outros. E para que eles obtenham sucesso, uma série de requisitos devem ser levados em consideração, como descreve (GRAY,1993 appud DURÃES,2007): como a relevância, que se refere a utilização de parâmetros reais da aplicação a ser avaliada pelo *benchmark*; portabilidade, onde o *benchmark* deve ser executado em diferentes arquiteturas computacionais e sob diferentes plataformas. O que possibilita a sua execução sob diferentes aplicações; escalabilidade, onde o *benchmark* pode ser executado tanto em pequenos como em super computadores, onde configurações e arquiteturas não devem influenciar na confiabilidade dos valores obtidos por ele e simplicidade de utilização da

4.2 Carga de trabalho

Como dito anteriormente, as ferramentas *benchmark* possibilitam a definição de transações que posteriormente são executadas para medir o desempenho do SGBDs. Muitas vezes, ligadas a operação de escrita e leitura dos dados implementadas em alguma linguagem de programação ou SQL. A essas transações dá-se o nome de carga de trabalho. Assim, a ferramenta mede o quanto o SGBD foi eficiente quando submetido a essa carga de trabalho específica.

A carga de trabalho deve refletir ao máximo os procedimentos e recursos utilizados na aplicação na hora de sua utilização, o que vem a dar credibilidade e realidade aos resultados adquiridos para análise de seu desempenho. Para a análise de banco de dados, Costerano (1997), especifica alguns parâmetros que devem ter uma atenção maior na hora da descrição das operações que constituem a carga de trabalho, como: o tipo da transação seja ela uma seleção ou atualização, os predicados utilizados como números e tipos, os valores finais e intermediários para os resultados das transações e a frequência relativa das transações.

4.3 BenchmarkSQL

O *benchmarkSQL* é uma ferramenta benchmark utilizada na análise de desempenho de banco de dados implementada em linguagem de programação Java e disponibilizada com código aberto. Seu desenvolvimento teve como base as implementações do TPC-C. Essa ferramenta faz utilização de driver JDBC para a comunicação do banco, o que o torna compatível com uma vasta gama de SGBDs bem como podendo ser utilizado tanto em sistemas operacionais Windows quanto Linux (LIMA, 2008).

A utilização dessa ferramenta benchmark, segundo (CARNEIRO; MOREIRA; FREITAS) concentra-se em quatro etapas principais, como: criação das tabelas, onde são criadas as tabelas do banco de dados que fazem parte do método TPC-C; carga das tabelas, onde é realizada a partir da execução do arquivo loadData.bat no Windows e load.Data.sh no Linux, gerando dados de forma aleatória nas tabelas criadas; criação de índices, onde são definidos os índices e as chaves que identificam cada tabela do banco de dados; execução da ferramenta. Após as três etapas anteriores, a ferramenta *benchmark* pode ser executada, para tal, é necessário informar o nome do arquivo de configuração do banco de dados utilizado. Com isso, a ferramenta se mostra a partir de uma interface gráfica semelhante a outras do sistema Windows, onde alguns parâmetros podem ser configurados conforme a necessidade.

5. Análise comparativa entre SGBDs PostgreSQL e Informix no acesso a base de dados de imagens

O trabalho aqui apresentado tem como principal objetivo realizar uma análise de desempenho entre os SGBDs PostgreSQL e Informix. Para alcançar tal objetivo, uma ferramenta *benchmark* contará com uma carga de trabalho composta por arquivos de imagens para realizar as avaliações. Por fim, será realizada uma comparação do desempenho entre ambos.

5.1 Procedimentos Metodológicos

O PostgreSQL versão 8.4.4 foi escolhido para a comparação pelo fato de ser um SGBD livre, ou seja, sem custos para sua utilização, e que emprega a utilização de modelos de dados objeto-relacional. Já o Informix Innovator se deu também pelo fato de possuir uma versão livre de banco de dados que também possibilita a utilização de modelos de dados do tipo objeto-relacional. Porém, o ponto forte de sua escolha se deu pelo fato de ser um produto da IBM que, com o levantamento do referencial bibliográfico, notou-se pouca utilização do mesmo em nível de pesquisas e desenvolvimentos científicos. O que nos faz entrar na dúvida de que a sua fraca utilização esta relacionada ao seu desempenho, ou pela cultura de utilizar produtos mais conhecidos.

A metodologia de comparação se deu pelo modelo experimental ou empírico, que é a análise do próprio sistema com a utilização de uma ferramenta *benchmark*, o

que torna a análise mais precisa e confiável. Assim, o banco de dados foi criado e os arquivos utilizados como carga de trabalho foram do tipo JPEG.

O *benchmark* utilizado foi o BenchmarkSQL 2.3.2, por ser uma ferramenta compatível com os bancos de dados utilizados nesse trabalho e com o SO (sistema operacional). Desenvolvido em *Java*, essa ferramenta é apresentada ao usuário através de interfaces gráfica semelhante a outras encontradas em ambiente Windows, facilitando sua configuração e utilização.

Todos os testes foram realizados utilizando uma mesma máquina, um computador *Notebook Toshiba Satellite* modelo L305-S5933, possuindo um processador Intel Pentium Dual Core com velocidade de 2.16 GHz, 1 MB em cachê, disco rígido SATA de 250 GB e 5400RPM (rotações por minuto) e 3 GB de memória RAM. O sistema operacional utilizado foi o GNU/Linux com distribuição UBUNTU 11.04, com o *kernel* versão 2.6.32 e com 1 GB de partição *swap*.

5.2 Testes

A ferramenta *benchmark* possibilita tanto a execução por quantidade de transações por terminais, quanto por tempo de execução, assim optou em realizar a execução em um tempo de 5 minutos cada, onde esse tempo deu como retorno uma quantidade média de 590.000 transações, sendo suficiente para a análise em questão. Enriquecendo ainda mais a análise, será executada uma carga de trabalho sobre os bancos de dados com quantidades diferentes de imagens, em análises com uma imagem, oito, dezesseis, trinta e duas, e sessenta e quatro imagens, ambas com imagens de 585KB e 1MB.

Visando resultado confiáveis, alguns cálculos estatístico foram realizados para a descoberta do numero de amostras mínimas a se realiza. O calculo principal é mostrado pela formula 1, onde (n_0) representa o número de amostras a ser realizada.

Para realizar o calculo, algumas variáveis são necessárias, como o valor de Z tabelado (Z_y), o valor da variância (σ^2) e valor da distancia máxima do valor populacional

(E_0) BARBETETTA; REIS; BORNIA, 2004).

$$n_0 = \frac{Z_y^2 \sigma^2}{E_0^2}$$

(1)

Descoberto todas essas variáveis, é possível calcular o tamanho das amostras para o PostgreSQL e para o Informix, conforme a formula 1 dada anteriormente. Como resultado, o número mínimo de amostras para o Informix é de 53 e para o PostgreSQL é de 72.

Com base nos resultados, serão executados 200 testes, sendo 100 para o PostgreSQL e 100 para o Informix, 50 com imagens de 582K com resolução 1024x1024, e outros 50 com imagens de 1MB e resolução de 1280x1024, onde, a cada 10 testes realizados, a quantidade de imagens utilizadas aumenta gradativamente.

5.3 Resultados

Com o término dos testes realizados através da ferramenta Benchmarksql nos bancos PostgreSQL e Informix, os resultados serão aqui apresentados. Como o número de amostras mínimas para a análise em questão foi significativamente alto, com o decorrer dos testes o número de imagens bem como o seu tamanho foi sendo elevado gradativamente, até que se completassem 100 replicações para cada banco, como relatado anteriormente.

Os primeiros resultados apresentados são referentes a 10 replicações com uma imagem de 582KB e mais 10 com uma imagem de 1MB. Como resultado, o Informix apresentou-se uma diferença de 1490,25 linhas por segundo a menos em relação ao resultado do PostgreSQL, obtendo um desempenho 24,74% inferior conforme mostra a figura 1.

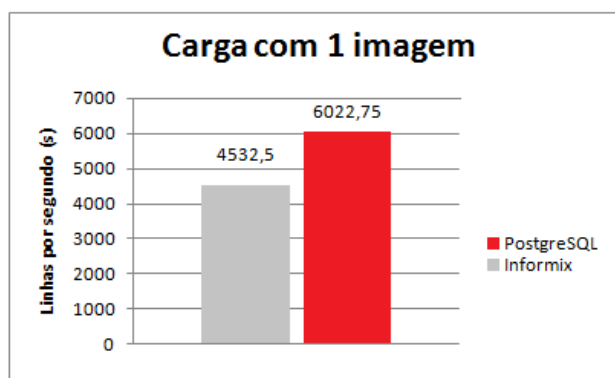


Figura 1. Gráfico comparativo do desempenho com uma imagem.

Em seguida a quantidade de imagens na carga de trabalho foi elevada, totalizando oito imagens iguais as das análises anteriores, assim outras 20 análises foram realizadas. O Informix manteve um desempenho inferior ao do PostgreSQL, como mostra a figura 2, uma diferença de 1468,55 linhas por segundo, 25,12%.

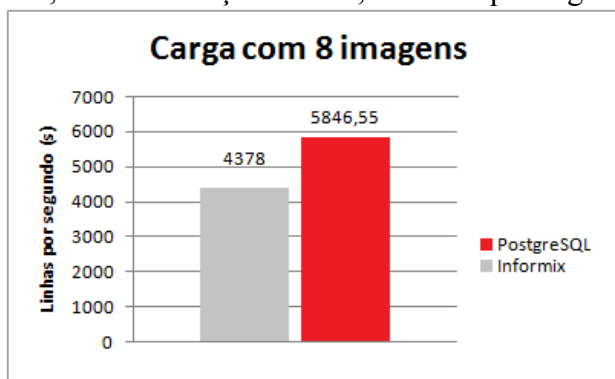


Figura 2. Gráfico comparativo de desempenho com carga de 8 imagens.

Com dezesseis imagens o Informix apresentou um desempenho muito semelhante aos resultados obtidos nas 20 análises anteriores. Já o PostgreSQL manteve uma queda de rendimento, como mostram as figuras 2 e 3.

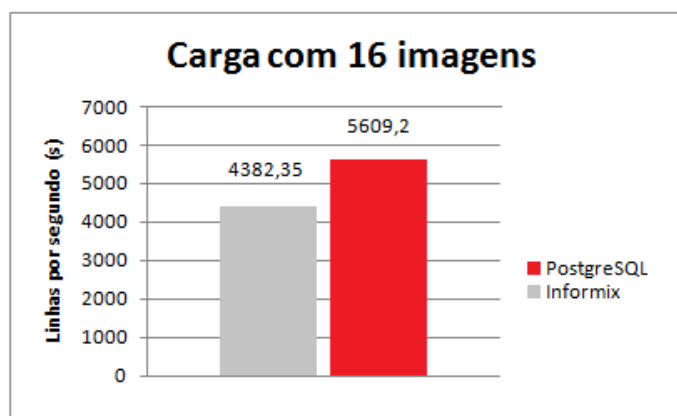


Figura 3. Gráfico comparativo de desempenho com carga de 16 imagens.

Dando continuidade a análise, novamente a carga de trabalho foi elevada, agora para trinta e duas imagens, onde novamente o Informix demonstrou uma pequena queda de rendimento se comparado com a queda de rendimento do PostgreSQL como mostram as figuras 3 e 4.

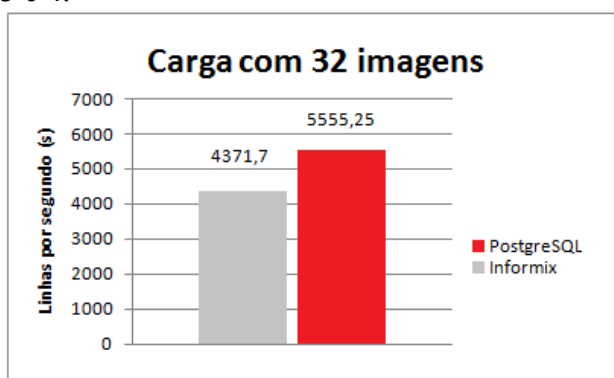


Figura 4. Gráfico comparativo de desempenho com carga de 32 imagens.

Neste caso específico, a diferença de rendimento do Informix em relação ao PostgreSQL foi de 23,30%. Finalizando o processo de análise de desempenho e totalizando as 100 replicações, os bancos de dados foram submetidos a uma carga de trabalho com sessenta e quatro imagens. Com isso os resultados demonstrados na figura 5 mostram que o Informix obteve um desempenho inferior, com exatas 1176,15 linhas processadas por segundo a menos que o PostgreSQL, uma diferença de 21,94%.

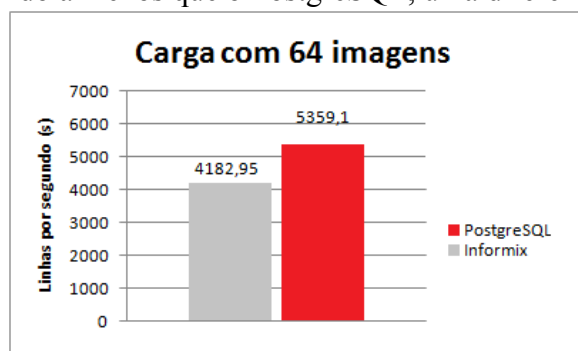


Figura 5. Gráfico comparativo de desempenho com carga de 64 imagens.

Por fim, pode-se notar que o PostgreSQL obteve um desempenho superior ao Informix, como mostra a figura 6. Porém, o mesmo foi o que apresentou maior queda de rendimento na medida em que a quantidade de imagens foi gradativamente aumentando,

apresentando uma queda de 11,01% se comparada com as primeiras 20 replicações. Com o Informix, na mesma situação, a queda foi de apenas 7,71%.

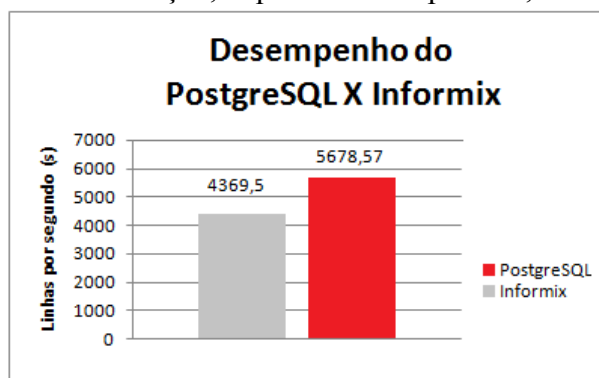


Figura 6. Comparação do desempenho do Informix X PostgreSQL.

Ao final, nota-se que a quantidade de amostras é fundamental para uma análise justa e confiável, onde se pode verificar que o resultado ao final de 100 replicações é consideravelmente diferente das primeiras 20 ou 40 replicações iniciais. Assim, o PostgreSQL obteve um desempenho 5678,57 linhas por segundo, contra 4369,5 do Informix, uma diferença final de 23,05%.

6. Conclusão

Através do presente estudo pode-se discorrer sobre conceitos e características dos SGBDs PostgreSQL e Informix, e de algumas ferramentas *benchmarks* tendo como principal foco o Benchmarksql. A carga de trabalho submetida aos bancos foi composta por imagens, que no decorrer dos testes, o número de imagens e o seu tamanho foram elevados gradativamente, enriquecendo ainda mais o resultado final.

Com o decorrer do desenvolvimento desta, o objetivo específico foi alcançado, que era a comparação de desempenho entre SGBDOR no acesso à base de dados de imagens por meio da utilização de uma ferramenta *benchmark*. Os objetivos específicos também foram sendo alcançados na medida em que o presente trabalho foi sendo desenvolvido, onde foram documentados conceitos referentes a bancos de dados e aos SGBDs utilizados, bem como a definição de possíveis ferramentas *benchmark* para a realização de avaliação de desempenho.

Em todos os casos analisados, o PostgreSQL obteve um desempenho superior ao do Informix, tendo um desempenho final 23,05% maior que o concorrente. Apesar disso, o Informix foi o SGBD que apresentou uma menor queda de rendimento na medida em que as amostras eram realizadas e o tamanho da carga era elevado. Com isso, esse trabalho pode servir como base para que desenvolvedores tomem a decisão de qual SGBD utilizar no desenvolvimento de uma aplicação, levando em consideração o tipo de informação manipulada.

Futuros trabalhos podem ser desenvolvidos utilizando diferentes tipos e formatos de arquivos multimídias, como áudio ou vídeo, de sistemas operacionais como o Windows ou GNU/Linux. Também pode ser avaliado o desempenho entre bancos de dados distintos, com o auxílio de diferentes ferramentas *benchmark*. Trabalhos esses que caminham todos com um bem comum, a avaliação e comparação de desempenho entre SGBDs.

7. Referencias

BARBETTA, Pedro Alberto; Reis, Marcelo Menezes; Borna, Antonio Cezar. **Estatística**: para cursos de engenharia e informática. São Paulo: Atlas, 2004.

BITTENCOURT, Danilo de Souza. Banco de dados de imagens: uma revisão da literatura. 2007. 57f. TCC (Ciência da Computação) – Universidade Federal de Lavras, 2007

CARVALHO, Adriano Arlei de. Fundamentação teórica para Processamento Digital de Imagens. 2003. 18 f. TCC (Ciência da Computação) – Universidade Federal de Lavras, Lavras, 2003

CIFERRI, Ricardo Rodrigues. Um benchmark voltado à análise de desempenho de sistemas de informações geográficas. Dissertação Mestrado (Ciência da Computação) – Universidade Estadual de Campinas, 1995.

COSTERANO, Walter Paulo. Caracterização de sistemas de banco de dados espaciais para análise de desempenho. Dissertação Mestrado (Ciência da Computação) – Universidade Estadual de Campinas, Campinas, 1997.

DATE, C. J. Introdução a sistemas de bancos de dados. 7. ed. Rio de Janeiro: Ed. Campus, 2000.

CARNEIRO, Alessandro P.; MOREIRA, Julinao L.; FREITAS, Andre L. C. Tuning – Técnicas de Otimização de Banco de Dados: Um Estudo Comparativo. Artigo. Universidade Federal do Rio Grande (UFRG), 2006.

DURÃES, Dayane Helen Santos. Análise de desempenho entre banco de dados Postgresql e Firebird na plataforma Windows 2000. 2007. 42 f. TCC (Sistemas de Informação) – Universidade Estadual de Monte Carlos, Monte Carlos, 2007.

ELMASRI, Ramez; NAVATHE, Sham. Sistemas de banco de dados: fundamentos e aplicações. Rio de Janeiro: LTC, 2002.

GONZÁLEZ, Sahudy Montenegro. Modelo para o Gerenciamento de Banco de Dados de Imagens. Tese Doutorado (Engenharia Elétrica) – Universidade Estadual de Campinas, 2003.

LIMA, Murilo Rodrigues de. Execução distribuída de benchmarks em sistemas de banco de dados relacional. Dissertação Mestrado (Ciência Exatas) – Universidade Federal do Paraná, Curitiba, 2008.

PEREIRA, Rafles Frausino. Análise de desempenho de banco de dados utilizando *benchmarks* especializados. 138 f. Dissertação de Mestrado (Ciência da Computação) – Universidade Estadual de Campinas, Campinas, 1990.

SILBERSCHATZ, Abraham; KORTH, Henry F.; SUDARSHAN, S. Sistema de banco de dados. 3. ed. São Paulo: Makron Books, 1999.

ANEXOS

Anexo 1. Distribuição *t* de Student.

gl	Área na cauda superior								
	0,25	0,10	0,05	0,025	0,01	0,005	0,0025	0,001	0,0005
1	1,000	3,078	6,314	12,71	31,82	63,66	127,3	318,3	636,6
2	0,816	1,886	2,920	4,303	6,965	9,925	14,09	22,33	31,60
3	0,765	1,638	2,353	3,182	4,541	5,841	7,453	10,21	12,92
4	0,741	1,533	2,132	2,776	3,747	4,604	5,598	7,173	8,610
5	0,727	1,476	2,015	2,571	3,365	4,032	4,773	5,894	6,869
6	0,718	1,440	1,943	2,447	3,143	3,707	4,317	5,208	5,959
7	0,711	1,415	1,895	2,365	2,998	3,499	4,029	4,785	5,408
8	0,706	1,397	1,860	2,306	2,896	3,355	3,833	4,501	5,041
9	0,703	1,383	1,833	2,262	2,821	3,250	3,690	4,297	4,781
10	0,700	1,372	1,812	2,228	2,764	3,169	3,581	4,144	4,587
11	0,697	1,363	1,796	2,201	2,718	3,106	3,497	4,025	4,437
12	0,695	1,356	1,782	2,179	2,681	3,055	3,428	3,930	4,318
13	0,694	1,350	1,771	2,160	2,650	3,012	3,372	3,852	4,221
14	0,692	1,345	1,761	2,145	2,624	2,977	3,326	3,787	4,140
15	0,691	1,341	1,753	2,131	2,602	2,947	3,286	3,733	4,073
16	0,690	1,337	1,746	2,120	2,583	2,921	3,252	3,686	4,015
17	0,689	1,333	1,740	2,110	2,567	2,898	3,222	3,646	3,965
18	0,688	1,330	1,734	2,101	2,552	2,878	3,197	3,610	3,922
19	0,688	1,328	1,729	2,093	2,539	2,861	3,174	3,579	3,883
20	0,687	1,325	1,725	2,086	2,528	2,845	3,153	3,552	3,850
21	0,686	1,323	1,721	2,080	2,518	2,831	3,135	3,527	3,819
22	0,686	1,321	1,717	2,074	2,508	2,819	3,119	3,505	3,792
23	0,685	1,319	1,714	2,069	2,500	2,807	3,104	3,485	3,768
24	0,685	1,318	1,711	2,064	2,492	2,797	3,091	3,467	3,745
25	0,684	1,316	1,708	2,060	2,485	2,787	3,078	3,450	3,725
26	0,684	1,315	1,706	2,056	2,479	2,779	3,067	3,435	3,707
27	0,684	1,314	1,703	2,052	2,473	2,771	3,057	3,421	3,689
28	0,683	1,313	1,701	2,048	2,467	2,763	3,047	3,408	3,674
29	0,683	1,311	1,699	2,045	2,462	2,756	3,038	3,396	3,660
30	0,683	1,310	1,697	2,042	2,457	2,750	3,030	3,385	3,646
35	0,682	1,306	1,690	2,030	2,438	2,724	2,996	3,340	3,591
40	0,681	1,303	1,684	2,021	2,423	2,704	2,971	3,307	3,551
45	0,680	1,301	1,679	2,014	2,412	2,690	2,952	3,281	3,520
50	0,679	1,299	1,676	2,009	2,403	2,678	2,937	3,261	3,496
z	0,674	1,282	1,645	1,960	2,326	2,576	2,807	3,090	3,291