UNIVERSIDADE FEDERAL RURAL DE PERNAMBUCO UNIDADE ACADÊMICA DE EDUCAÇÃO A DISTÂNCIA E TECNOLOGIA BACHARELADO EM SISTEMAS DE INFORMAÇÃO  
BACHARELADO EM SISTEMAS DE INFORMAÇÃO

**ALEXSANDRO MATIAS DE ALMEIDA**

**ANÁLISE DE TESTE DE CARGA EM DIFERENTES BANCOS DE DADOS RELACIONAIS UTILIZANDO BENCHMARK PADRÃO DE SUPORTE À DECISÃO**

PALMARES, 2020

**ALEXSANDRO MATIAS DE ALMEIDA**

**ANÁLISE DE TESTE DE CARGA EM DIFERENTES BANCOS DE DADOS RELACIONAIS UTILIZANDO BENCHMARK PADRÃO DE SUPORTE À DECISÃO**

Trabalho de Conclusão de Curso apresentado à Coordenação do Curso de Bacharelado em Sistemas de informação, pelo o aluno ALEXSANDRO MATIAS DE ALMEIDA, sob orientação do professor Clayton Carvalho da Trindade, para conclusão do Curso de Sistemas de Informação.

PALMARES, 2020

**AGRADECIMENTOS**

Agradeço, primeiramente, a Deus pela oportunidade dessa formação e por ter me dado saúde e força para superar as dificuldades.

A minha família por todo amor, paciência, incentivo e apoio incondicional que me deram durante a minha jornada no curso.

E a todos que direta ou indiretamente fizeram parte da minha formação, o meu muito obrigado.

**RESUMO**

O presente estudo analisa o desempenho de diferentes Bancos de Dados Relacionais utilizando benchmark TPC-H que serve na medição de carga de trabalho que auxilia de suporte à decisão. Esse parâmetro deve ser mensurado, uma vez que é um dos fatores bastante relevante na seleção de um banco de dados. Desta forma, o teste de performance será realizado em um ambiente virtualizado para garantir que para o teste seja aplicado em condições pré-determinadas e devidamente replicadas de formas mais idênticas e controladas possíveis em ambos ambientes.

**Palavras-Chave:** Bancos de Dados Relacionais, Desempenho, TPC-H.

**ABSTRACT**

The present study analyzes or performs different Related Databases using virtualization methods. This parameter must be measured, since it is one of the most relevant factors in the selection of a database. In this way, the performance test will be carried out in a virtualized environment to ensure that the test is applied under pre-applied conditions and properly replicated in the most identical and controlled ways possible in both environments.

**Keywords:** Relacional Databases, Performance.

**LISTA DE SIGLAS**

TPC-H – Transaction Processing Performance Council

ACID – Atomicidade, Consistência, Isolamento e Durabilidade.

BD – Banco de Dados.

DDL - Data Definition Language.

DML - Data Manipulation Language.

SGBD – Sistema de Gerenciamento de Banco de Dados.

SGBDR – Sistema de Gerenciamento de Banco de Dados Relacional.

SQL – Structured Query Language.

MER – Modelo Entidade Relacionamento

**Sumário**

[1. INTRODUÇÃO 8](#_Toc45006124)

[1.1 OBJETIVOS 8](#_Toc45006125)

[1.1.1 OBJETIVO GERAL 8](#_Toc45006126)

[1.1.2 OBJETIVOS ESPECÍFICOS 8](#_Toc45006127)

[1.2 JUSTIFICATIVA 9](#_Toc45006128)

[2. FUNDAMENTAÇÃO TEÓRICA 9](#_Toc45006129)

[2.1 SISTEMA DE GERENCIAMENTO DE BANCO DE DADOS 10](#_Toc45006130)

[2.1.1 BANCO DE DADOS MYSQL 10](#_Toc45006131)

[2.1.2 BANCO DE DADOS POSGRESQL 11](#_Toc45006132)

[2.2 COMANDOS SQL 11](#_Toc45006133)

[2.2.1 DDL DATA DEFINITION LANGUAGE 12](#_Toc45006134)

[2.2.2 DML DATA MANIPULATION LANGUAGE 12](#_Toc45006135)

[2.2.3 DCL DATA CONTROL LANGUAGE 12](#_Toc45006136)

[2.3 CONTAINERS 12](#_Toc45006137)

[2.4 PERFORMANCE 13](#_Toc45006138)

[2.4.1 TESTE DE CARGA OU VOLUME 13](#_Toc45006139)

[2.4.2 TESTE DE CAPACIDADE 13](#_Toc45006140)

[3. PROCEDIMENTOS METODOLÓGICOS 13](#_Toc45006141)

[3.1 METODOLOGIA 13](#_Toc45006142)

[3.2 TIPO DE PESQUISA 14](#_Toc45006143)

[3.3 AMOSTRA E UNIVERSO 14](#_Toc45006144)

[3.4 COLETA DOS DADOS 14](#_Toc45006145)

[3.5 HARDWARE UTILIZADO 14](#_Toc45006146)

[4. ANÁLISE DOS DADOS DA PESQUISA 15](#_Toc45006147)

[5. CONSIDERAÇÕES FINAIS 15](#_Toc45006148)

[6. REFERÊNCIAS 16](#_Toc45006149)

1. INTRODUÇÃO

A concepção dos bancos de dados relacionais é indispensável em sistemas de informação para aplicações no campo empresarial. Isso se torna visível quando cinco dos bancos dados mais utilizados no mercado nos últimos cinco anos, quatro deles são relacionais. Estes são representados em ordem crescente em Oracle, MySQL, Microsoft SQL Server, PostgreSQL e MongoDB (DB-Engines, 2020). Para este estudo será adotado Sistemas de Gerenciamento de Banco de Dados (SGBD) gratuitos, uma vez que existe a Cláusula DeWittde que proíbe pesquisadores e cientistas divulgarem explicitamente os nomes de seus sistemas em trabalhos acadêmicos (Brian Moran, 2003).

Então, já que os bancos MySQL e PostgreSQL estão entre os cinco mais utilizados no mercado (DB-Engines, 2020), se torna interessante conhecimento mais específico quanto à performance dos mesmos no que diz respeito ao tempo gasto nas operações de inserção, busca, alteração e exclusão de dados, uso de CPU e utilização de memória RAM ao aplicar uma carga de dados no sistema.

1.1 OBJETIVOS

* + 1. OBJETIVO GERAL

Analisar a performance de bancos de dados relacionais através de testes de carga utilizando o benchmark padrão TPC-H.

* + 1. OBJETIVOS ESPECÍFICOS
* Realizar teste de carga nos bancos de dados MySQL e PostgreSQL;
* Desenvolver uma metodologia capaz de realizar um teste de carga de forma crescente de registros determinada pelo método de benchmark;
* Analisar os resultados obtidos.
  1. JUSTIFICATIVA

O aumento crescente da utilização dos bancos de dados é atribuído aos avanços nas tecnologias de sistemas de informação que consideram os dados das empresas como um bem extremamente valioso. Essas ferramentas têm auxiliado as empresas a guiar o seu planejamento estratégico. Assim, com o crescimento exponencial desses dados, a disponibilidade, integridade, confidencialidade e forma de armazenamento são imprescindíveis neste processo de competitividade empresarial. Outra aplicação direta do uso de banco de dados é o conceito de Big Data que vem se consolidando como a base para o desenvolvimento de novas tecnologias capazes de manipular grandes conjuntos de dados visando também promover subsídios para a tomada de decisão. Dito isto, algumas características importantes na implementação desses bancos de dados para o mercado:

* Performance aceitável na manipulação a grandes volumes de dados;
* Flexibilizar a atual estrutura baseada em modelo de dados e metadados utilizada pelos bancos relacionais;
* Escalabilidade do modelo do banco de dados implementado.

O estudo da performance de banco de dados é de relevante importância, uma vez que se lida com um recurso intangível tão valioso.

Este trabalho terá na seção 2 serão apresentados os conceitos básicos, caraterísticas em comum e aplicação desses bancos de dados, assim como será explanado o ambiente virtualizado aonde foram realizados os testes. Na seção 3 é apresentada a metodologia e preparação do ambiente virtualizado para o teste de carga, apresentando quais os softwares (com suas respectivas versões) e hardware foram utilizados, além de tratar de algumas métricas preestabelecidas juntamente com as configurações especificas dos bancos de dados estudados. Já na seção 4 serão mostrados os resultados obtidos no teste de carga, para que se possa concluir qual dos dois qual deles é o mais eficiente.

1. FUNDAMENTAÇÃO TEÓRICA

Nesta seção são abordados conceitos básicos necessários para a compreensão dos principais temas abordados nesta pesquisa. De uma forma geral serão explanados os conceitos fundamentais em torno de Sistemas de Gerenciamento de Banco de Dados, os comandos SQL e suas particularidades. Os conceitos em torno dos testes necessários para avaliação de desempenho dos dois Banco de Dados em estudo e testes estatísticos utilizados na análise dos resultados.

* 1. dados
  2. informação
  3. SISTEMA DE GERENCIAMENTO DE BANCO DE DADOS

Originalmente da sigla em inglês DBMS (Data Base Management System), um Sistemas de Gerenciamento de banco de dados (SGBD), ou é um pacote de software projetado cuja função é gerenciar uma base dados. As vantagens do seu uso são:

* Independência de Dados: Os programas aplicativos não devem, idealmente, ser expostos aos detalhes de representação e armazenamento de dados.
* Acesso Eficiente aos Dados: Um SGBD utiliza uma variedade de técnicas sofisticadas para armazenar e recuperar dados eficientemente.
* Integridade e Segurança dos Dados: Se os dados são sempre acessados através do SGBD, ele pode forçar restrições de integridade.
* Administração de Dados: Quando diversos usuários compartilham dados, centralizar a administração dos dados pode oferecer melhorias significativas.
* Acesso Concorrente e Recuperação de Falha: Um SGBD planeja o acesso concorrente aos dados de maneira tal que os usuários podem achar que os dados estão sendo acessados por apenas um único usuário de cada vez. Além disso, o SGBD protege os usuários dos efeitos de falhas de sistema.
* Tempo Reduzido de Desenvolvimento de Softwares: O SGBD suporta funções importantes que são comuns a várias linguagens de programação que acessam os dados no SGBD.

É importante ressaltar (Ramakrishnan, et al., 2009) que algumas vezes, não se torna viável o uso de um SGBD é um software complexo, otimizado para alguns tipos de processamento (por exemplo, responder a consultas complexas ou tratar várias requisições concorrentes), e seu desempenho pode não ser adequado para determinados aplicações especificas. Entretanto, na maioria das situações em que é necessário gerenciamento de dados em grande escala, os SGBDs têm se tornado uma ferramenta indispensável.

* + 1. BANCO DE DADOS MYSQL

O MySQL é um sistema de gerenciamento de banco de dados (SGBD), que utiliza a linguagem SQL (Linguagem de Consulta Estruturada, do inglês Structured Query Language) como interface. É atualmente um dos sistemas de gerenciamento de bancos de dados mais populares da Oracle Corporation, com mais de 10 milhões de instalações pelo mundo.

Possui as seguintes características:

* É um Software Livre com base na GPL;
* Alta portabilidade já que suporta praticamente qualquer plataforma atual;
* Boa Compatibilidade com linguagens de programação pois existe drivers ODBC, JDBC e .NET e módulos de interface para diversas linguagens de programação, como Delphi, Java, C/C++, C#, Visual Basic, Python, Perl, PHP, ASP e Ruby);
* Excelente desempenho e estabilidade;
* Pouco exigente quanto a recursos de novos hardwares;
* Facilidade no manuseio;
* Contempla a utilização de vários Storage Engines como MyISAM, InnoDB, Falcon, BDB, Archive, Federated, CSV, Solid;
* Suporta controle transacional;
* Suporta Triggers;
* Suporta Cursors (Non-Scrollable e Non-Updatable);
* Suporta Stored Procedures e Functions;
* Replicação facilmente configurável;
  1. COMANDOS SQL

Structured Query Language, ou em tradução livre, Linguagem Estruturada de Consulta é a linguagem usada nos SGBDs por padrão, no entanto, cada um tem suas particularidades dentro da própria linguagem, tendo implementações diferentes (Atanazio, 2019). A linguagem SQL tem algumas divisões, que facilitam o entendimento da mesma, categorizando seus comandos. Sendo que as mais conhecidas, que serão explicadas a seguir, são: DDL, DML e DCL.

* + 1. DDL DATA DEFINITION LANGUAGE

Linguagem de Definição de Dados, é a parte da Linguagem SQL que trata, como o próprio nome diz, da definição da estrutura dos dados, cujos efeitos se dão sobre objetos. Esses comandos são utilizados para a criação de bancos de dados, tabelas, views, triggers. Exemplos de comandos: CREATE (criação), ALTER (alteração), DROP (remoção), etc.

* + 1. DML DATA MANIPULATION LANGUAGE

Linguagem de Manipulação de Dados, é a parte da Linguagem SQL que não altera a estrutura e sim os registros de uma base de dados, cujos efeitos se dão sobre registros (Atanazio, 2019). São comandos que fazem consultas, inserem, alteram ou apagam registros. Exemplos de comandos: SELECT (consulta), INSERT (inserção), UPDATE (alteração), DELETE (remoção), etc.

* + 1. DCL DATA CONTROL LANGUAGE

Linguagem de Controle de Dados, é a parte da linguagem SQL referente ao controle de acesso a objetos por usuários e seus respectivos privilégios. Os principais comandos são:

* GRANT: Garante (permite) acesso dado a um usuário;
* REVOKE: Revoga (retira) direitos dados a um usuário. Os direitos dados a um usuário podem ser: ALL, CREATE, EXECUTE, REFERENCES, SELECT, TRIGGER, USAGE, CONNECT, DELETE, INSERT, RULE, TEMPORARY, UPDATE, etc.
  1. PERFORMANCE

Em tecnologia, refere-se ao desempenho da aplicação, o tempo de resposta para executar uma operação de forma satisfatória, garantindo também que a aplicação seja escalável (IZAC, 2018).

* + 1. TESTE DE CARGA OU VOLUME

É utilizado para identificar o limite da aplicação, qual o máximo de processamento que pode suportar trabalhando com cargas acima do normal, seja para quantidade de acessos simultâneos ou quantidade de operações\transações processadas com grande volume.

Para isso, antes dos testes começarem, define-se quais métricas serão coletadas e analisadas. A mais comum é o throughput, que é a taxa de transferência do processamento. Por exemplo, quantos registros foram processados por segundo, conceito usado bastante para redes, mas existe também o tempo de resposta, consumo de CPU e memória. É preciso conhecer essas métricas em condições normais para efeitos de comparação e análise das métricas quando o processamento validado for acima do normal.

* + 1. TESTE DE CAPACIDADE

O teste de capacidade medirá as questões listadas nos testes de carga citado acima, porém considera um intervalo de tempo pré-determinado para essas medições.

1. PROCEDIMENTOS METODOLÓGICOS
   1. METODOLOGIA

A metodologia de elaboração deste trabalho está dividida nas seguintes etapas:

* Etapa 1: Elaboração do Modelo Entidade Relacionamento (MER)
* Etapa 2: Instalação e configuração dos bancos de dados
* Etapa 3: Criação e população das tabelas;
* Etapa 4: Pesquisa de metodologias utilizadas em de trabalhos científicos relacionados ao estudo de performance de banco de dados.
* Etapa 5: Aplicados os testes.
* Etapa 6: Coleta e análise de resultados.

Para que seja viabilizado o estudo, deve-se tomar como início pelos seguintes parâmetros:

* Sistema operacional utilizado;
* Qual o banco de dados utilizado;
* Versão do banco utilizado
* Presença e ausência de índices;
* Qual mecanismo de criação de tabelas (engine) utilizado;

Números de registros variando em três possibilidades: 1.000, 10.000 e 50.000 registros.

* 1. TIPO DE PESQUISA
  2. AMOSTRA E UNIVERSO
  3. COLETA DOS DADOS
  4. HARDWARE UTILIZADO

1. ANÁLISE DOS DADOS DA PESQUISA
2. CONSIDERAÇÕES FINAIS
3. REFERÊNCIAS

Patibandla. Pavan. **How a single PostgreSQL config change improved slow query performance by 50x**. 2017. Disponível em: <https://amplitude.engineering/how-a-single-postgresql-config-change-improved-slow-query-performance-by-50x-85593b8991b0>. Acesso em: 15 de Junho 2020.

leopard blog. PostgreSQL Indexes. 2015. Disponível em: <https://leopard.in.ua/2015/04/13/postgresql-indexes#.Xt7c2NrPyUl>. Acesso em: 15 de Junho 2020

Ramakrishnan Raghu e Gehrke Johannes. Sistemas de Gerenciamento Sistemas de Gerenciamento de Banco de Dados [Livro]. - São Paulo : McGraw-Hill, 2009.

Schwartz Baron, Zaitsev Peter e Tkachenko Vadim. High Performance MySQL: Optimization, Backups, and Replication. 2ª. ed. Massachusetts : O’Reilly, 2012.

Vincent Arel-Bundock. List of Datasets. 2020. Disponível em: <https://vincentarelbundock.github.io/rdatasets/articles/data.html>. Acesso em: 01 de Junho 2020.

Micro Focus. Teste de carga: A chave para manter os aplicativos de negócios em execução. 2017. Disponível em:

DB-Engines. DB-Engines Ranking. 2020. Disponível em: <https://db-engines.com/en/ranking>. Acesso em: 14 de Junho 2020.

JIN, Xiaolong; WAH, Benjamin W.; CHENG, Xueqi; WANG, Yuanzhuo. Significance and Challenges of Big Data Research. Big Data Research, n.2, p. 59-64. 2015.

Izac, Ariane. Introdução a Testes de Performance. 2018. Disponível em: < http://www.matera.com/blog/post/introducao-a-testes-de-performance>. Acessado em: 15 de Junho 2020.

wikipedia. David DeWitt. 2020. Disponível em: <https://en.wikipedia.org/wiki/David\_DeWitt>. Acessado em: 15 de Junho de 2020.

Moran, Brian. The Devil's in the DeWitt Clause**.** 2003. Disponível em: https://www.itprotoday.com/sql-server/devils-dewitt-clause. Acessado em: 16 de Junho de 2020