

Uma Abordagem Experimental para Avaliar o Desempenho do Banco de Dados Open-Source PostgreSQL

Herderson Couto¹, Francisco Airton Silva², Gustavo Callou¹, Ermeson Andrade¹

¹ Universidade Federal Rural de Pernambuco (UFRPE) - Recife, PE – Brasil

² Universidade Federal do Piauí (UFPI) - Picos, PI – Brasil

{herderson.couto, gustavo.callou, ermeson.andrade}@ufrpe.br, faps@ufpi.edu.br

Abstract. *A Database Management System (DBMS) is a software structure responsible for storing the data of an application. In these systems, is also possible to organize the data so that it can be added, updated, deleted and accessed whenever necessary. Databases are essential today and are present in virtually all types of applications. Due to its wide range of use, several DBMSs were created to meet existing demands. Although the most used databases today are the relational type, they still need updated experimental studies that bring more clarification about their performance, even in high demand situations. In this work, we present an experimental approach to evaluate PostgreSQL performance, in order to understand how this database behaves in different usage situations, focusing mainly on the evaluation of response time. The results show how the increase in the number of simultaneous users, requests per second and workload inserted or queried can negatively influence PostgreSQL performance, leading the database to significant performance losses.*

Resumo. *Um Sistema de Gerenciamento de Banco de Dados (SGBD) é uma estrutura de software responsável por armazenar os dados de uma aplicação. Nesses sistemas, também é possível organizar os dados de modo que possam ser acrescentados, atualizados, excluídos e acessados sempre que for necessário. Os bancos de dados são essenciais atualmente e estão presentes em praticamente todos os tipos de aplicações. Pela sua ampla gama de utilização, vários SGBDs foram criados para atender as demandas existentes. Muito embora os bancos de dados mais utilizados atualmente são os do tipo relacional, os mesmos ainda necessitam de estudos experimentais atualizados que tragam mais esclarecimentos sobre o seu desempenho, inclusive em situações de alta demanda. Neste trabalho, apresentamos uma abordagem experimental para avaliar o desempenho do PostgreSQL, com a finalidade de compreender como esse banco de dados se comporta diante de situações variadas de uso, focando principalmente na avaliação do tempo de resposta. Os resultados mostram como o aumento da quantidade de usuários simultâneos, de requisições por segundo e da carga de trabalho inserida ou consultada podem influenciar negativamente no desempenho do PostgreSQL, levando o banco de dados a perdas significativas de performance.*

1. Introdução

A necessidade de armazenamento de dados é algo que vem crescendo ano após ano. Ao passo que a humanidade evolui na era digital, o volume de informações geradas cresce ex-

ponencialmente. Estudos apontam uma crescente no volume de informações digitais geradas [Gantz and Reinsel 2012]. Alinhados a esse crescimento, os SGBDs, vêm evoluindo para viabilizar o armazenamento e o processamento dessa grande massa de informação. Vários são os bancos de dados disponíveis para uso no mercado, porém os mais populares são os bancos relacionais, que ainda hoje ocupam as primeiras posições no ranqueamento dos SGBDs mais usados no mundo [DB-Engines 2022]. Por outro lado, os bancos de dados NoSQL (*Not only SQL*), vêm ganhando popularidade devido a sua capacidade de lidar com grandes volumes de dados e possibilitando o armazenamento de informações em *schemas* flexíveis, tais como no formato de grafo ou documento, por exemplo [Candel et al. 2022].

O banco de dados é fundamental na maioria das aplicações atualmente, visto que uma má configuração ou um gargalo nesse componente pode resultar em um baixo desempenho da aplicação. Dentre os bancos de dados existentes, o modelo relacional se destaca pela simplicidade e flexibilidade, onde os dados são representados e armazenados em relações/tabelas [Machado 2020]. Porém, esse modelo possui limitações quando é preciso levar em conta a heterogeneidade do cenário de Big Data atual e a necessidade de escalabilidade horizontal [Hsieh et al. 2017]. Isso não quer dizer que eles não sejam mais úteis, mas quando é necessário levar em consideração o grande volume de informações, velocidade e a necessidade de modelos de dados flexíveis, os bancos NoSQL são mais indicados. Desde a década de 90 é possível encontrar na literatura técnica trabalhos que avaliaram o desempenho do PostgreSQL. Todavia, a avaliação dos bancos relacionais considerando diferentes operações e cargas de trabalho continua sendo extremamente importante. Novas versões do PostgreSQL vem sendo lançadas no decorrer dos anos e estudos experimentais atualizados que avaliam os limites de desempenho e recursos dessas versões trazem uma boa contribuição ao meio científico. Além disso, podem ajudar a melhor dimensionar servidores de banco de dados.

No estado da arte, encontramos trabalhos com estudos voltados para a análise de desempenho em bancos de dados. Em [Dopke et al. 2021], os bancos PostgreSQL, MySQL e MongoDB tiveram seu desempenho avaliado em um ambiente de Nuvem Privada, considerando cargas relacionadas ao sequenciamento de DNA. Em [Makris et al. 2021], um estudo comparativo sobre aspectos de desempenho foi realizado com os bancos de dados MongoDB e PostgreSQL, a fim de identificar qual dos dois bancos é mais eficiente em termos de tempo de resposta dentro de cenários específicos. Já em [Lima et al. 2018], um comparativo de desempenho entre o PostgreSQL e o Cassandra foi realizado para demonstrar qual banco de dados se sairia melhor em um cenário em que uma grande quantidade de dados precisava ser migrada. Porém, nenhum desses trabalhos focam em fazer uma análise abrangente do desempenho do PostgreSQL considerando, por exemplo diferentes cargas de trabalho e operações com vários usuários simultâneos.

Neste trabalho, apresentamos uma abordagem baseada em experimentos para avaliar o SGBD PostgreSQL. Para isso adotamos um ambiente real para executar os experimentos de fatorial completo [Cheng 2016] visando analisar os *trade-offs*, considerando por exemplo, diferentes operações e cargas de trabalho aplicadas ao PostgreSQL. Os resultados mostram que quando inserimos ou consultamos poucos dados (50, 100 e 200 linhas), as médias de tempo de resposta não apresentam uma diferença expressiva umas das outras. Os resultados também revelam que a operação de escrita demanda mais tempo

de processamento do que a leitura dos dados, resultando em tempos de resposta maiores. Este trabalho pode auxiliar os desenvolvedores de aplicações ou indivíduos a decidir qual configuração é melhor para a demanda do ambiente de trabalho.

O restante deste artigo está organizado como segue. A Seção 2 apresenta a fundamentação do trabalho, onde os principais conceitos são introduzidos. A Seção 3 mostra os trabalhos relacionados. A Seção 4 apresenta a arquitetura experimental. A Seção 5 detalha os resultados obtidos. Finalmente, a Seção 6 apresenta as conclusões e descreve brevemente os trabalhos futuros.

2. Fundamentação

Os Sistemas de Gerenciamento de Banco de Dados Relacionais (SGBDR) estão presentes na maioria das aplicações e sistemas atualmente. Apesar do grande crescimento no uso dos bancos NoSQL, os SGBDR seguem sendo bastante utilizados devido as suas ótimas características de integridade e consistência de dados. Vale destacar que os bancos de dados relacionais são bancos baseados no modelo relacional de dados [Date 2004]. Este modelo foi criado pelo matemático britânico Edgar Frank Codd em 1970, quando era pesquisador no laboratório da IBM [Codd 1983].

No modelo relacional de banco de dados, os dados são armazenados em relações/tabelas, as quais podem armazenar, por exemplo, informações relacionadas a clientes como nome, telefone, etc. As tabelas são compostas por tuplas e atributos. Nas tuplas/linhas são encontradas as informações de uma determinada ocorrência em uma tabela. Os atributos/colunas armazenam as características de um determinado registro, tais como nome ou CPF de um determinado cliente. Também existe a possibilidade de expressar relacionamentos, representando como as tabelas estão associadas umas com as outras. Vale destacar que o modelo relacional padronizou mundialmente a forma de armazenar informações em banco de dados, tendo o SQL (*Structured Query Language*) como a linguagem declarativa padrão para os SGBDRs. Apesar de existirem vários bancos de dados, este trabalho adotou o PostgreSQL devido ao fato do mesmo ser um dos bancos de dados mais usados no mundo [DB-Engines 2022].

3. Trabalhos Relacionados

Existem disponíveis na literatura vários trabalhos que focam na análise de desempenho de banco de dados. Em [Pires et al. 2006], um comparativo entre os dois bancos de dados relacionais de código aberto mais populares do mundo (MySQL e PostgreSQL), obteve através de um *benchmark*, métricas necessárias para fazer uma comparação de desempenho entre os dois SGBDs. Mais uma comparação de performance entre banco de dados foi feita em [Politowski and Maran 2014], onde os bancos utilizados nesta comparação foram o MongoDB e o PostgreSQL. Os testes foram divididos em três categorias: inserção, busca simples e busca complexa. Os resultados apontaram que o MongoDB obteve os melhores resultados, baseado no tempo de execução das operações de leitura e escrita.

Em [Sharma et al. 2018], os autores realizaram uma comparação de desempenho dos banco de dados PostgreSQL, MongoDB e Neo4J. A comparação envolveu uma base de dados do *OpenStreetMap* para avaliar o desempenho dos três bancos de dados através de consultas simples, a fim de comparar o desempenho por meio do tempo de resposta. Em [Medeiros et al. 2020], os autores utilizaram VMs da Google Cloud Platform (GCP)

para criar o ambiente de teste e avaliar a replicação de dados no SGBD relacional MySQL, com o foco em estratégias de Recuperação de Desastres (RD). Foi levado em consideração vários cenários de testes utilizando o JMeter. O trabalho mostrou que métricas como o tempo de resposta, o Recovery Point Objective (RPO) e o Recovery Time Objective (RTO) variam de acordo com o tamanho dos dados replicados, o tipo de sincronização e a configuração das réplicas no MySQL.

Outro trabalho que avaliou o desempenho em bancos de dados foi em [Klimek and Skublewska-Paszowska 2021]. Nesse trabalho, os autores compararam o desempenho do PostgreSQL e do MySQL com base em um aplicativo escrito em Java. A partir dos resultados obtidos, os autores concluíram que o PostgreSQL é mais eficiente em baixa carga de trabalho, já em casos de recuperação de dados, o MySQL obteve os melhores resultados. Em [Santos et al. 2019], foi realizada uma avaliação de desempenho baseada em análise de sensibilidade de contêineres Docker executando os SGBDs relacionais MySQL, PostgreSQL e SQLServer. O trabalho teve como objetivo principal identificar qual o banco de dados relacional deve ser utilizado junto ao Docker a fim de obter o melhor desempenho.

Este trabalho apresenta algumas semelhanças e diferenças com os trabalhos acima citados. Apesar dos trabalhos listados anteriormente trabalharem na linha de avaliação de desempenho de um ou mais banco de dados, nenhum explorou de forma profunda o tempo de resposta, considerando diferentes fatores na arquitetura experimental como diferentes intervalos de requisições, número de usuários e operações (ex: escrita e leitura). Além disso, nenhum desses trabalhos mostrou os pontos de saturação dos bancos avaliados, revelando, por exemplo, a taxa de erro. Nem tão pouco chegaram a monitorar os recursos do servidor no decorrer de algum teste para verificar os efeitos da carga de trabalho no hardware. A Tabela 1 resume as comparações.

Tabela1. Trabalhos Relacionados.

Trabalho Relacionado	Tipo de SGBD	Arquitetura cliente-servidor	Teste de estresse	Monitoramento de recursos
[Pires et al. 2006]	Relacional	Não	Não	Não
[Politowski and Maran 2014]	Relacional e Não Relacional	Não	Não	Não
[Sharma et al. 2018]	Relacional e Não Relacional	Não	Não	Não
[Medeiros et al. 2020]	Relacional	Sim	Não	Não
[Klimek and Skublewska-Paszowska 2021]	Relacional	Não	Não	Não
[Santos et al. 2019]	Relacional	Sim	Não	Não
O presente trabalho	Relacional	Sim	Sim	Sim

4. Arquitetura Experimental

Esta seção detalha a arquitetura experimental usada para a realização dos experimentos.

4.1. Ambiente de Testes

Para realizar os testes de performance, foi montada uma estrutura com dois computadores, sendo um cliente e o outro funcionando como servidor de banco de dados. Nessa estrutura, o cliente envia requisições ao banco de dados instalado no servidor, que prontamente responde a tais solicitações. Os dois computadores foram conectados através de uma rede local cabeada (Fast Ethernet), onde somente as duas máquinas estavam conectadas a rede. A vantagem de usar uma rede cabeada e isolada do tráfego de pacotes de outros hosts é a diminuição de possíveis interferências externas nos resultados obtidos. Deste modo, podemos obter resultados mais fidedignos quanto ao comportamento do PostgreSQL, diante de uma situação de estresse.

A máquina cliente teve as seguintes configurações de hardware: Processador Intel Core I5 3230M 2,6 GHz, com 6 GB de memória RAM e SSD de 240 GB. O sistema operacional instalado foi o Windows 10, 64 bits. O computador designado para ser o servidor de banco de dados teve as seguintes configurações de hardware: Processador Intel Core I5 480 M 2,66 GHz, memória RAM de 6 GB e HD de 500 GB. Foi instalado o sistema operacional Ubuntu 20.04.2 de 64 bits. O elemento de rede que conectou as duas máquinas foi um roteador de rede convencional, modelo L1-RW131 da fabricante Link1One. A versão do PostgreSQL instalada no servidor foi a 13.3, onde as configurações padrão do sistema foram todas mantidas.

4.2. Execução dos Experimentos e Medições

Para o envio de requisições de forma automatizada, foi utilizado o Apache JMeter versão 5.4.1. O JMeter é um software de código aberto escrito em Java projetado para realização de testes e medição de desempenho em diferentes tipos de sistemas (aplicativos, servidores, protocolos) [JMeter™ 2022]. Com essa ferramenta, é possível realizar todos os testes, considerando a quantidade de usuários simultâneos, os intervalos entre as requisições e a variação de carga de trabalho recebida pelo SGBD. O JMeter também nos permitiu coletar os dados com as informações necessárias para todas as análises.

O teste consistiu em enviar requisições do tipo INSERT e SELECT a uma tabela criada no PostgreSQL por um tempo de 300 segundos para cada grupo de usuários executado no JMeter. Entre cada grupo de usuários executado, foi respeitado um tempo de repouso para que o sistema operacional e o SGBD restabelecessem as condições normais de operação. Com a execução do teste, coletamos o tempo de resposta das requisições durante o tempo desejado. Vale salientar que o tempo de resposta é uma métrica de desempenho responsável por informar o tempo que um sistema leva para dar o retorno desejado ao usuário. No PostgreSQL, criamos uma tabela com três colunas: uma coluna configurada como chave-primária e com um auto incremento para armazenar o ID de cada dado inserido, e mais duas colunas do tipo *varchar* para armazenar o nome e o sobrenome, respectivamente, de um dado inserido.

Para os testes de leitura, foram inseridos previamente dois mil registros na tabela criada, com a finalidade de cobrir toda a carga a ser consultada nos testes, a cláusula LIMIT foi utilizada para poder limitar a quantidade de tuplas retornadas em uma consulta, conforme o parâmetro de linhas retornadas adotado em um teste. Já nos testes de inserção, os dados foram inseridos na tabela através do JMeter e a cada rodada de teste a tabela era zerada para evitar que o grande volume de dados inseridos interferisse no desempenho do banco de dados. Se considerarmos o tempo de execução de forma ininterrupta de todos os testes feitos neste trabalho, tanto o de escrita como o de leitura e sem considerar os tempos de pausa entre a execução de um grupo de usuários e outro, temos um total de mais de 28 horas de execução de experimentos.

4.3. Cenários Analisados

Vários parâmetros foram considerados para realização dos testes de performance com o PostgreSQL. Na Tabela 2, podemos ver todas as variações de parâmetros que fizeram parte do conjunto de testes. Foram consideradas as operações de leitura e escrita, intervalos entre as requisições que variaram entre 0,1 e 2 segundos, número de usuários

simultâneos acessando o banco de dados entre 5 e 100 usuários e linhas retornadas ou inseridas entre 50 e 2000 linhas. Ao todo, foram executados 336 testes para obter os dados necessários a esta análise e cobrir todos os cenários possíveis da Tabela 2. Os dados dos testes podem ser encontrados em ¹.

Tabela2. Parâmetros usados nos testes.

Operação	Intervalo entre as requisições (seg)	Número de Usuários	Linhas Consultadas ou Inseridas por Amostra
Leitura	0,1	5	50
Escrita	0,5	10	100
		20	200
	2	30	500
		50	1000
		80	2000
		100	

5. Resultados Experimentais e Discussão

Nesta seção, serão apresentados os resultados que foram obtidos através da execução dos experimentos.

5.1. Resultados dos Cenários Adotados

Após coletar todos os dados, geramos representações visuais dos dados de modo a interpretar e compreender o comportamento do PostgreSQL diante das configurações propostas com os parâmetros da Tabela 2. Vale destacar que adotamos o experimento fatorial completo [Cheng 2016], assim todas as combinações de testes com os parâmetros da Tabela 2 foram realizadas.

Na Figura 1, temos todos os resultados obtidos referentes aos teste que envolve-ram a operação de leitura. Analisando o comportamento desses gráficos, podemos notar que quando a consulta envolve poucas linhas retornadas (50, 100 e 200 linhas), não temos uma diferença expressiva no tempo de resposta, mesmo quando a quantidade de usuários simultâneos cresce. Isso mostra a capacidade do PostgreSQL em processar tais solicitações sem efeitos significativos na performance do ambiente. Já em cargas maiores (500, 1000 e 2000 linhas), fica clara a distância que essas linhas tomam uma das outras, principalmente quando usamos um intervalo entre as requisições menores (0,1 e 0,5).

Também podemos notar que quando certa quantidade de usuários são considerados, picos no tempo de resposta são visíveis. Esse comportamento é nítido na Figura 1 c), com a linha que representa a consulta de 2000 linhas (linha rosa). Nesta configuração, percebemos uma diferença expressiva no tempo de resposta médio, saindo de 62,1 ms com 50 usuários simultâneos para 232,9 ms com 80 usuários simultâneos. Outra fato que podemos destacar nesses gráficos é a diminuição do tempo de resposta, quando temos intervalos entre as requisições maiores. Quando comparamos os resultados em que o intervalo entre as requisições foi de 0,1 segundos (ver Figura 1 a)) com os resultados em que o intervalo entre as requisições foi de 2 segundos (ver Figura 1 d)), percebemos que as médias do tempo de resposta é muito menor no segundo caso. Isso acontece porque quanto maior o intervalo entre as requisições, menor é a quantidade de requisições que chegam ao sistema, o que reduz a carga de trabalho do SGBD na hora de processar as informações. Várias outras análises podem ser feitas. Por exemplo, a diferença média entre os intervalos entre as requisições de 0,1 e 2 (ver Figura 1 a) e d)), onde a quantidade de

¹Disponível em: <https://www.dropbox.com/sh/zx6mpcht6jz78b1/AAB2uh9sRd5FAiMq28ZLp3u1a?dl=0>

usuários é de 100 e a quantidade de linhas consultadas é de 2000, foi de aproximadamente 821,6 ms.

Uma análise estatística foi realizada com o intuito de verificar se os tempos de respostas de cada gráfico da Figura 1 são estatisticamente iguais ou diferentes entre si. Para verificar tal informação, foi utilizado o teste não paramétrico de Kruskal-Wallis [McKight and Najab 2010] que é capaz de verificar através do valor de p (p -value) se as medianas populacionais são iguais ou não. Como o valor de p obtido para todas as comparações foi menor que 0,05 (5%), é possível dizer que os tempos de respostas são estatisticamente diferentes entre si.

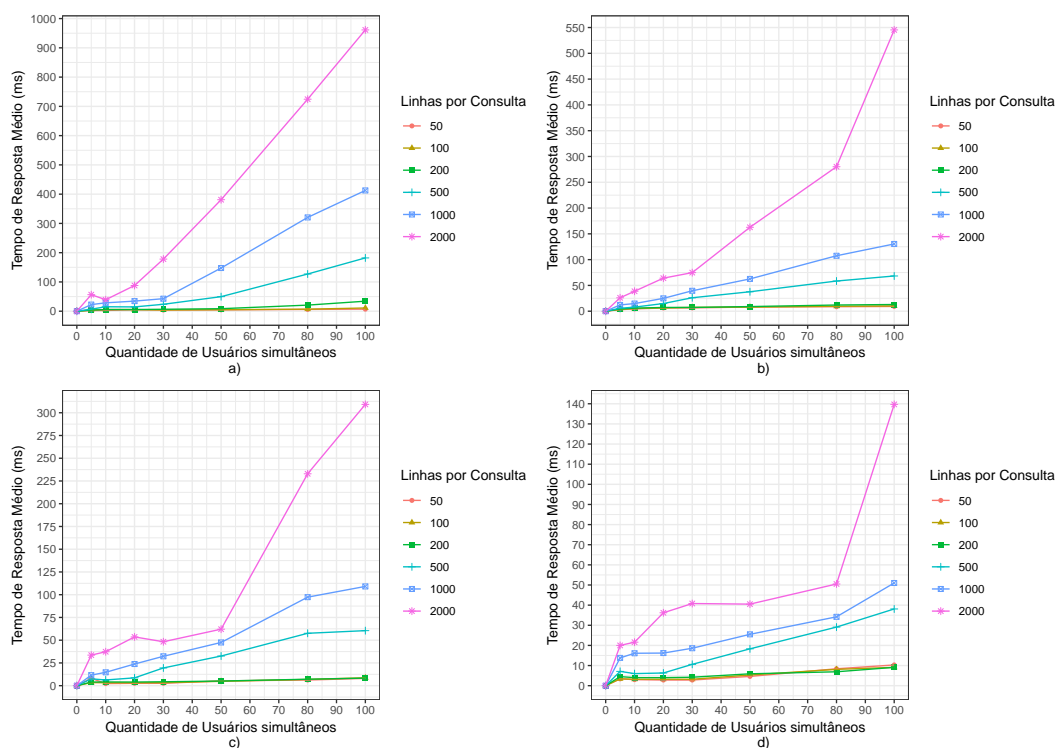


Figura 1. Tempo de resposta médio vs. Quantidade de usuários simultâneos - Leitura de Dados (SELECT). a) Intervalo entre as requisições de 0,1 (s); b) Intervalo entre as requisições de 0,5 (s); c) Intervalo entre as requisições de 1 (s); e d) Intervalo entre as requisições de 2 (s).

Na Figura 2, temos os gráficos referentes aos testes de inserção de dados. Se compararmos as médias dos tempos de respostas dos gráficos da Figura 1 com os da Figura 2, conseguimos constatar que o PostgreSQL leva muito mais tempo para inserir os dados em uma tabela do que para buscá-los. Com a análise dos dados, também conseguimos identificar alguns comportamentos que se repetiram nos gráficos da Figura 1 e Figura 2. A diferença de tempo só é maior quando temos grandes quantidades de usuários. As linhas só ficam mais espaçadas quando o número de usuários simultâneos cresce e as inserções são iguais ou maiores que 500 linhas. O tempo de resposta é menor quando temos intervalos entre as requisições maiores. Por exemplo, na Figura 2 d) o maior tempo atingido foi de 837,3 ms, inserindo 2000 linhas com 100 usuários simultâneos, já na Figura 2 a) o maior tempo atingido foi de 1972,3 ms, o que corrobora tal comportamento.

Podemos observar também na Figura 2 que não temos uma diferença significativa de desempenho quando a quantidade de usuários simultâneos é pequena, somente quando essa quantidade passa de 20 usuários é que notamos uma maior diferença. Ou seja, o PostgreSQL começa a perder desempenho a partir de 30 usuários simultâneos. Algo que podemos destacar também é que tivemos menos picos na evolução nas linhas dos gráficos de escrita, o que mostra um crescimento mais gradual do tempo com esse tipo de operação. Ambos os resultados das Figuras 1 e 2 mostram os pontos onde o SGBD começa a sofrer perda de desempenho e em quais situações ele apresenta uma melhora na performance. Ficou claro também que em ambas as operações (leitura e escrita) a quantidade de usuário simultâneos interferiu de maneira negativa no tempo de resposta que o banco de dados retorna ao usuário, além da alta quantidade de requisições por segundo. Também realizamos o teste estatístico de Kruskal-Wallis para verificar se os tempos de respostas para os gráficos com os testes de escrita são iguais ou não. O valor de p para esses dados também foi abaixo de 0,05 (5%), o que mostra que os tempos de respostas são estatisticamente diferentes entre si.

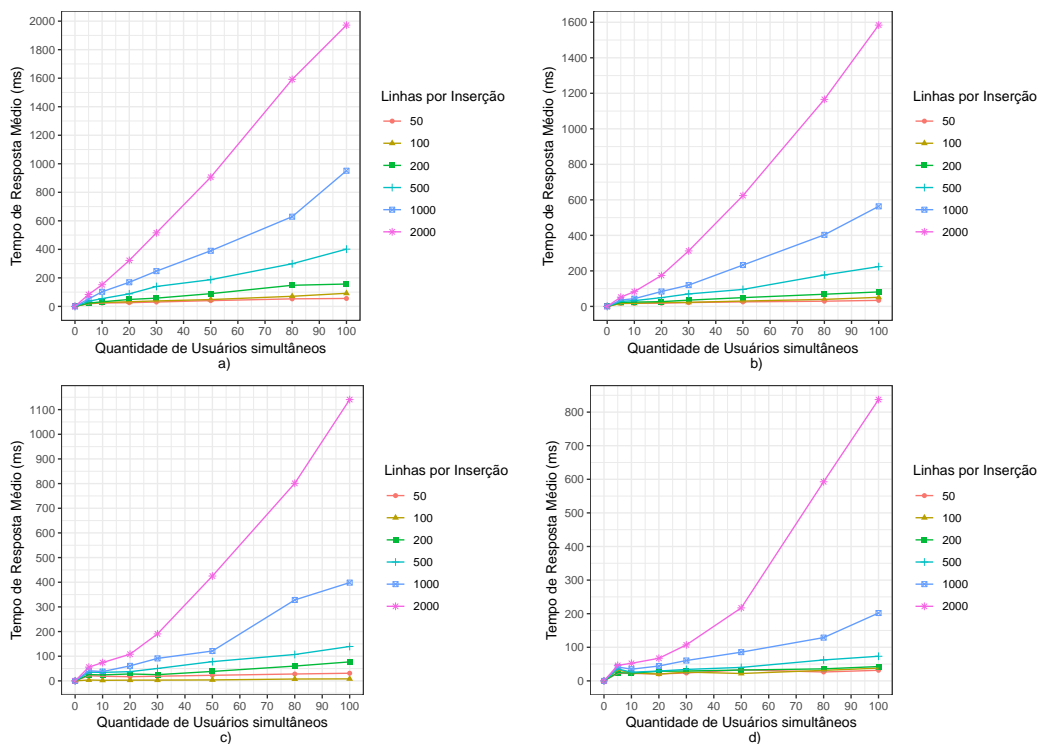


Figura 2. Tempo de resposta médio vs. Quantidade de usuários simultâneos - Escrita dos Dados (INSERT). a) Intervalo entre as requisições de 0,1 (s); b) Intervalo entre as requisições de 0,5 (s); c) Intervalo entre as requisições de 1 (s); e d) Intervalo entre as requisições de 2 (s).

5.2. Teste de Estresse

A fim de encontrar o limite de carga suportada pelo PostgreSQL nas configurações padrão, foram realizados testes de estresse para sobrecarregá-lo e encontrar gargalos que afetam o desempenho do SGBD. Para isso, foram utilizados os parâmetros que apresentaram a maior carga de trabalho para o banco de dados. Ou seja, o intervalo entre as requisições de

0,1 segundos e o número de linhas inseridas ou consultadas igual a 2000 linhas, foram os parâmetros que apresentaram os maiores tempos de resposta. Portanto, esses parâmetros foram escolhidos para os testes de estresse. Além dos dois parâmetros anteriores, a quantidade de usuários simultâneos foi variada de 25 a 200 usuários, a fim de verificar o desempenho do PostgreSQL com mais de 100 conexões simultâneas.

Na Figura 3, temos o gráfico com os resultados dos testes de estresse considerando a operação de leitura dos dados. No eixo Y primário temos os tempos de resposta médio, enquanto no eixo Y secundário temos a taxa de erro do teste (linha vermelha). Já no eixo X temos a quantidade de usuários de 25 a 200. Observando os resultados da Figura 3, percebemos que o PostgreSQL apresentou falhas ao responder as requisições quando a quantidade de usuários simultâneos foi superior a 100. Com 125 usuários simultâneos, a taxa de erro foi de 29,19%, ou seja, do total de requisições enviadas, 29,19% delas falharam. Isso mostra que o banco de dados começou a perder confiabilidade na hora de responder as requisições do usuário. O percentual de falhas nas requisições aumentou ainda mais quando a quantidade de usuários simultâneos cresceu, chegando a uma taxa de erro de 59,91% com 200 conexões simultâneas ao servidor de banco de dados.

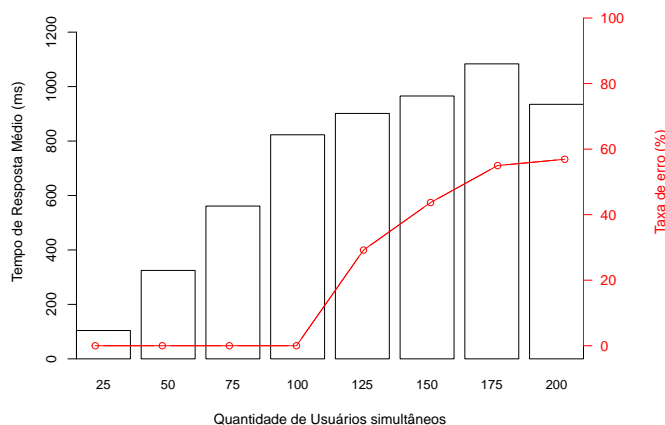


Figura 3. Teste de estresse de leitura - Tempo de Resposta Médio e Taxa de Erro vs. Quantidade de Usuários Simultâneos.

Na Figura 4, temos os resultados considerando a escrita. Nesses resultados, assim como nos de leitura, não tivemos perda de requisições até 100 usuários simultâneos, porém as taxas de erro foram consideravelmente mais elevadas, sendo todas acima de 76%, o que poderia inviabilizar o uso do PostgreSQL. Analisando as Figuras 3 e 4, podemos concluir que além de 100 usuários simultâneos o banco de dados pode resultar em uma perda considerável de transações. Para melhorar o desempenho do PostgreSQL seria necessário modificar as configurações padrão do sistema, desde que sejam tomados os devidos cuidados com o dimensionamento do hardware do servidor.

Uma vez que a operação de escrita retornou os maiores tempos de resposta, realizamos o monitoramento dos recursos de hardware para essa operação. Na Figura 5, temos o comportamento da CPU e da memória RAM ao longo da execução do teste de estresse relacionado a operação de escrita. O uso da CPU chegou a 100% próximo dos 50 minutos, justamente no momento em que temos mais usuários simultâneos realizando

as requisições ao PostgreSQL. A memória RAM, por outro lado, teve o seu uso mais elevado, porém estável ao longo da execução do experimento. Ela teve um uso médio de 87,46%, chegando ao uso máximo de 98,08%. Vale destacar que existem algumas quedas de uso dos recursos de CPU e de memória RAM. Isso ocorreu devido ao intervalo dado entre as execuções dos grupos de usuários.

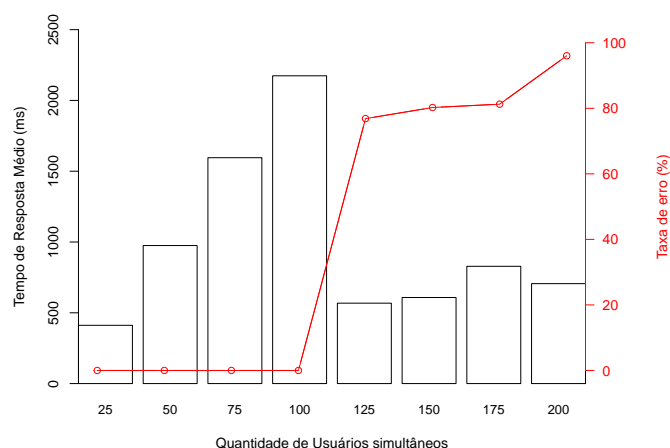


Figura 4. Teste de estresse de escrita - Tempo de Resposta Médio e Taxa de Erro vs. Quantidade de Usuários Simultâneos.

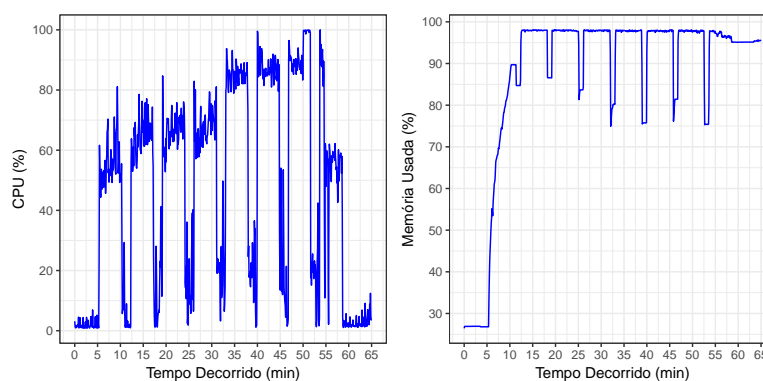


Figura 5. Monitoramento da CPU e da Memória RAM durante o teste de estresse relacionado a operação de Escrita.

5.3. Ameaças à validade

As ameaças à validade deste trabalho são descritas a seguir.

Generalidade do estudo experimental. Como qualquer estudo experimental, nossos resultados podem não ser facilmente generalizados para outros ambientes de bancos de dados relacionais ou versões do PostgreSQL. No entanto, acreditamos que o ambiente adotado possa ser encontrado facilmente em pequenas empresas ao redor do mundo. Além disso, o procedimento de análise mostrado neste trabalho pode ser facilmente adaptado a outros bancos ou versões do PostgreSQL.

Limitação da duração e do número de execuções experimentais. Experimentos de performance são uma atividade demorada e sua duração é inevitavelmente limitada como em qualquer trabalho experimental. Embora tenhamos realizado 336 testes para representar todos os cenários possíveis da Tabela 2, não podemos garantir que todas as tendências de degradação de performance sejam detectadas, pois algumas delas podem ter um tempo maior de degradação (e.x.: envelhecimento de software).

Representatividade da carga de trabalho adotada. A carga de trabalho adotada faz o uso do JMeter para o envio de requisições. Assim, é possível ao leitor obter resultados diferentes dos nossos, se for considerada uma carga de trabalho diferente. Além disso, as cargas de trabalho consideradas em nosso estudo considera os intervalos entre 0,1 e 2 segundos e o número de usuários simultâneos entre 5 e 100 que podem não representar comportamentos reais do usuário. Como o objetivo deste trabalho é expor potenciais tendências de performance, limitamos nosso estudo a essas cargas de trabalho.

6. Conclusões

Neste trabalho, realizamos testes de performance com a finalidade de avaliar o comportamento do PostgreSQL em diversas situações. Os testes submetem o SGBD a diversas cargas de trabalho dentro de uma estrutura cliente-servidor, colocando o sistema em situações de estresse em determinadas combinações de parâmetros de teste. A partir dos resultados obtidos, conseguimos extrair o conhecimento necessário para fazer uma leitura do desempenho do PostgreSQL. O nosso estudo experimental apresentou resultados relevantes que dão uma visão de como o PostgreSQL pode se comportar perante uma situação de alta demanda de usuários e requisições por segundo, por exemplo.

A partir dos resultados podemos concluir que a operação de escrita nos retorna um tempo maior de resposta do que a leitura dos dados e que intervalos maiores entre as requisições resultam em tempos de respostas menores. Também fica nítido que quanto maior a quantidade de usuários utilizando o BD, maior é a perda de desempenho. Os testes de estresse revelaram que o PostgreSQL começa a retornar falhas nas requisições quando a quantidade de usuários simultâneos passa de 100. Os resultados obtidos neste trabalho, pode de auxiliar profissionais tanto da área de banco de dados, como desenvolvedores de softwares e afins a escolher as melhores configurações do PostgreSQL com mais segurança, evitando problemas futuros de desempenho nos seus projetos. Como trabalhos futuros pretendemos considerar outras infraestruturas computacionais, como uma nuvem pública com máquinas virtuais a fim de analisar os *trade-offs* do PostgreSQL em tal ambiente.

Referências

- Candel, C. J. F., Ruiz, D. S., and García-Molina, J. J. (2022). A unified metamodel for nosql and relational databases. *Information Systems*, 104:101898.
- Cheng, C.-S. (2016). *Theory of Factorial Design*. Chapman and Hall/CRC.
- Codd, E. F. (1983). A relational model of data for large shared data banks. *Communications of the ACM*, 26(1):64–69.
- Date, C. J. (2004). *Introdução a sistemas de bancos de dados*. Elsevier Brasil.

- DB-Engines (2022). DB-Engines Ranking. <https://db-engines.com/en/ranking>. [Online; accessed 12-ago-2022].
- Dopke, L., Rockenbach, D., and Griebler, D. (2021). Avaliação de desempenho para banco de dados com genoma em nuvem privada. In *Anais da XXI Escola Regional de Alto Desempenho da Região Sul*, pages 45–48. SBC.
- Gantz, J. and Reinsel, D. (2012). The digital universe in 2020: Big data, bigger digital shadows, and biggest growth in the far east. *IDC iView: IDC Analyze the future*, 2007(2012):1–16.
- Hsieh, M.-J., Ho, L.-Y., Wu, J.-J., and Liu, P. (2017). Data partition optimisation for column-family nosql databases. *International Journal of Big Data Intelligence*, 4(4):263–275.
- JMeter™, A. (2022). Apache JMeter - What can I do with it? <https://jmeter.apache.org/>. [Online; accessed 04-fev-2022].
- Klimek, B. and Skublewska-Paszkowska, M. (2021). Comparison of the performance of relational databases postgresql and mysql for desktop application. *Journal of Computer Sciences Institute*, 18:61–66.
- Lima, A. T. C., Diógenes, V. F., and Pereira, J. Q. (2018). Estudo comparativo entre postgresql (sgbd) e apache cassandra (nosql). *Anais do Encontro de Computação do Oeste Potiguar ECOP/UFERSA (ISSN 2526-7574)*, (2).
- Machado, F. N. R. (2020). *Banco de Dados–Projeto e Implementação*. Saraiva Educação SA.
- Makris, A., Tserpes, K., Spiliopoulos, G., Zisis, D., and Anagnostopoulos, D. (2021). MongoDB vs postgresql: A comparative study on performance aspects. *GeoInformatica*, 25(2):243–268.
- McKight, P. E. and Najab, J. (2010). Kruskal-wallis test. *The corsini encyclopedia of psychology*, pages 1–1.
- Medeiros, W., Mendonça, J., Alves, G., and Andrade, E. (2020). Avaliação experimental de replicação em banco de dados para recuperação de desastres. In *Anais do XIX Workshop em Desempenho de Sistemas Computacionais e de Comunicação*, pages 121–132. SBC.
- Pires, C. E., Nascimento, R. O., and Salgado, A. C. (2006). Comparativo de desempenho entre bancos de dados de código aberto. *Escola Regional de Banco de Dados, Anais da ERBD06, Porto Alegre*.
- Politowski, C. and Maran, V. (2014). Comparação de performance entre postgresql e mongodb. *X Escola Regional de Banco de Dados*. SBC, pages 1–10.
- Santos, B., Endo, P. T., and Silva, F. A. (2019). Uma avaliação de desempenho de contêineres docker executando diferentes sgbds relacionais. In *Anais do XVIII Workshop em Desempenho de Sistemas Computacionais e de Comunicação*. SBC.
- Sharma, M., Sharma, V. D., and Bundele, M. M. (2018). Performance analysis of rdbms and no sql databases: Postgresql, mongodb and neo4j. In *2018 3rd International Conference and Workshops on Recent Advances and Innovations in Engineering (ICRAIE)*, pages 1–5. IEEE.