





Analysis of Software Aging in a Database Environment

Herderson Couto , Ermeson Andrade , Francisco Airton Silva , Gustavo Callou 

Abstract—Computer systems that run for long periods of time can suffer from a phenomenon known as software aging. Just like people, software can age. This phenomenon, however, can be viewed as a problem in computer systems because it can accelerate the depletion of resources and even lead to system failures. Databases are widely used software nowadays and can be affected by such a phenomenon, since they need to run for long periods of time uninterruptedly. Therefore, studies that investigate the possible effects of software aging in database environments are very necessary. In this work, we experimentally investigate the software aging phenomena in a database environment using PostgreSQL as the DBMS (Database Management System). By performing statistical analysis on the measurement data, we detected a suspicious phenomenon of software aging induced by workloads in memory and CPU usage. Additionally, our process analysis identified suspicious processes that can lead to memory degradations.

Index Terms—software aging, PostgreSQL, database, statistical analysis

I. INTRODUÇÃO

Os bancos de dados são softwares essenciais nos dias de hoje. Eles estão presentes na maioria das aplicações e podem exercer um papel importantíssimo em uma organização, permitindo o armazenamento e o acesso a uma grande variedade de informações. Existem vários SGBDs (Sistemas de Gerenciamento de Banco de Dados) disponíveis no mercado, sendo os do tipo relacional os mais utilizados [1].

Os sistemas de gerenciamento de banco de dados são softwares projetados para ter uma alta disponibilidade [2], tendo que funcionar, muitas vezes, de modo ininterrupto. Esses sistemas também costumam processar uma alta quantidade de transações por segundo, e por isso, problemas com potencial de causar interrupções inesperadas precisam ser detectados e tratados para que não causem prejuízos aos usuários desses sistemas. Alguns SGBDs se destacam no mundo por sua popularidade, sendo um dos mais populares o PostgreSQL, que conquistou uma forte reputação por apresentar robustez, confiabilidade e desempenho consistente.

Herderson Couto, Departamento de Computação, Universidade Federal Rural de Pernambuco, Brasil, herderson.couto@ufrpe.br

Ermeson Andrade, Departamento de Computação, Universidade Federal Rural de Pernambuco, Brasil, ermeson.andrade@ufrpe.br

Francisco Airton Silva, Universidade Federal do Piauí, Brasil, faps@ufpi.edu.br

Gustavo Callou, Departamento de Computação, Universidade Federal Rural de Pernambuco, Brasil, gustavo.callou@ufrpe.br

Qualquer software está sujeito a *bugs*. Eles podem ocorrer ao longo da utilização dos softwares e causar algum tipo de prejuízo aos seus usuários, como o completo travamento do sistema ou a lentidão [3]. Esses sistemas podem ser acometidos por um fenômeno conhecido como "envelhecimento de software", sendo ele responsável pela deterioração gradual do sistema ao longo do tempo [4].

Um dos sintomas mais comuns do envelhecimento de software é o vazamento de memória [5]. A presença de envelhecimento em um ambiente de execução de um SGBD pode afetar negativamente o seu desempenho, causando por exemplo, aumento do tempo de resposta ou a ocorrência de falhas que gerem riscos de interrupções. Apesar de existirem trabalhos que analisam o envelhecimento de software em bancos de dados [6], experimentos de longa duração não foram realizados e as origens do problema não foram investigadas.

Neste trabalho, foi investigado experimentalmente os sintomas de envelhecimento de software em um ambiente de banco de dados usando o PostgreSQL como SGBD, implantado em um sistema operacional Linux. Nos experimentos, foram realizadas consultas sucessivas a tabelas previamente criadas no banco de dados, utilizando diferentes intervalos entre as requisições para as consultas e diferentes quantidades de usuários simultâneos. Também foram coletadas várias métricas de desempenho do sistema, como uso de memória RAM, de SWAP e de CPU, e aplicado o teste de Mann-Kendall [7] [8] para confirmar as suspeitas de envelhecimento de software no ambiente com o PostgreSQL. Além disso, foram identificados os processos suspeitos da causa do envelhecimento do software. Nossas descobertas, bem como o método de análise, podem ser úteis para usuários ou desenvolvedores no processo de tomada de decisão para neutralizar, por meio de monitoramento e ajustes contínuos, problemas de envelhecimento de software encontrados nesses ambientes.

O restante deste artigo está organizado como segue. A Seção 2 mostra os trabalhos relacionados. A Seção 3 apresenta a arquitetura experimental. A Seção 4 detalha os resultados obtidos. Finalmente, a Seção 5 apresenta as conclusões e descreve brevemente os trabalhos futuros.

II. TRABALHOS RELACIONADOS

Na literatura, existem vários trabalhos que analisaram o envelhecimento de software em ambientes computacionais. Em Andrade [9], um estudo experimental investigou o potencial efeito de envelhecimento de software em um sistema de classificação de imagem executado em um ambiente de computação em nuvem. Os mesmos autores em [10], também

investigaram os efeitos de envelhecimento de software em sistemas de classificação de imagem. Porém, realizando um comparativo dos efeitos em ambientes de nuvem e borda.

Já em plataformas *Blockchain*, existem alguns trabalhos que fazem análises de envelhecimento de software. Dias [11] realizaram uma análise para investigar os eventos que afetam e provocam o envelhecimento de software na plataforma de *Blockchain* Cardano. Os autores em [12] avaliaram a plataforma de *Blockchain* Hyperledger Fabric, de modo a encontrar vazamento de recursos. Também existem trabalhos que estudaram o envelhecimento em contêineres. Os trabalhos [13] e [14] fizeram uma investigação de sintomas de envelhecimento de software na plataforma de contêiner Docker. Adicionalmente, alguns trabalhos visaram detectar sintomas de envelhecimento de software no sistema operacional Android. No trabalho de Huo [15], foi utilizado algoritmos de *machine learning* para detectar o envelhecimento de software no sistema Android. Já no trabalho [16], foi realizado um estudo empírico sobre manifestações de *bugs* relacionados ao envelhecimento de software no Android.

São escassos na literatura trabalhos que buscaram investigar a presença do fenômeno de envelhecimento de software em ambientes de banco de dados. Uma exceção é o trabalho [6]. Nesse trabalho, os autores investigaram os efeitos do envelhecimento de software causados pela ativação de *bugs* no MySQL. Diferente do trabalho citado, neste artigo foi investigada a presença do envelhecimento de software em um ambiente de banco de dados utilizando o PostgreSQL como SGBD. Além disso, foram monitorados os processos do sistema operacional e os do banco dados a fim de identificar os processos que tiveram uma maior influência na degradação da memória ao longo do tempo. Adicionalmente, foram executados os experimentos de longa duração e comprovada a suspeita de envelhecimento de software por meio do teste de Mann-Kendall.

III. PLANO EXPERIMENTAL

Esta seção detalha o plano experimental usado para a realização dos experimentos.

A. Ambiente de Testes

Para realização dos experimentos, foi montada uma estrutura com dois computadores, um cliente e o outro sendo utilizado como um servidor de banco de dados. Nessa estrutura, o cliente envia consultas SQL ao banco de dados, que responde prontamente às solicitações do cliente. As duas máquinas foram conectadas por meio de uma rede local cabeada (*Fast Ethernet*). Somente as duas máquinas ficaram conectadas no decorrer dos testes para garantir um tráfego de rede isolado, sem pacotes de *hosts* diferentes. Com isso, é possível obter resultados mais precisos (sem ruído) quanto ao comportamento do servidor de banco de dados, diante das cargas de trabalho utilizadas.

O componente de rede que conectou os dois computadores foi um roteador de rede convencional modelo L1-RW131 da fabricante Link1One. O SGBD utilizado no servidor foi o PostgreSQL versão 14.4. Nas duas máquinas foi utilizado o

sistema operacional Linux Ubuntu versão 20.04 de 64 bits, as configurações de *hardware* do computador cliente e do servidor são descritas na Tabela I.

TABELA I
CONFIGURAÇÃO DE HARDWARE DOS COMPUTADORES UTILIZADOS.

Computador	Hardware
Cliente	Processador Intel Core i7-4790 3,6 GHz, HD de 1 TB, Memória RAM de 8 GB
Servidor	Processador AMD A10 pro 7800b r7, HD de 250 GB, Memória RAM de 7 GB

B. Execução dos Experimentos

De modo ininterrupto por um período de 48 horas, para cada uma das cargas de trabalho adotada, foi executado o experimento que consistiu no envio de consultas SQL pela máquina cliente à máquina designada como servidor de banco de dados. Cada consulta realizada pelo cliente retornou um total de quatro mil linhas com o uso do comando SQL a seguir:

```
SELECT C.ID_cliente , C.Nome_Cliente ,
C.Sobrenome_cliente , Pr.nome_produto ,
Pr.preco_produto , P.qtde FROM
Clientes AS C JOIN Pedidos AS P
ON C.ID_cliente = P.ID_cliente
JOIN Produtos AS Pr ON
Pr.ID_produto = P.ID_produto ;
```

Os parâmetros das cargas usados na Tabela II foram escolhidos no intuito de acelerar o aparecimento de potenciais sintomas de envelhecimento de software no ambiente do servidor [17]. É importante salientar que o servidor foi reiniciado sempre ao fim da execução de cada carga de trabalho, ou seja, executou-se a carga baixa durante 48 horas, reiniciou-se o servidor. Posteriormente, executou-se a carga média. E assim sucessivamente. Com a reinicialização do sistema operacional, é possível garantir a normalização no uso de recursos de hardware e software do servidor para o início de um novo teste com uma nova carga, removendo possíveis acúmulos de efeitos causados pelo envelhecimento de software. Além disso, limpar as áreas de cache.

TABELA II
PARÂMETROS USADOS NOS TESTES - CONSULTAS
RETORNANDO 4000 TUPLAS.

Carga	Intervalo entre as requisições (seg)	Número de usuários simultâneos
Baixa	1	5
Média	0,5	50
Alta	0,2	100

Na Figura 1 é apresentado o *schema* de banco de dados adotado para o experimento. Ele é composto por três tabelas, com 4000 tuplas de registros inseridos em cada uma delas, estando elas relacionadas pelas chaves estrangeiras na tabela "pedidos", nenhuma estrutura de índice foi criada para otimizar

as consultas. Note que a finalidade deste trabalho foi verificar se com o envio constante das cargas de trabalho adotadas na Tabela II, ia ser possível detectar por meio da análise dos dados de monitoramento do uso da memória RAM, da CPU e dos processos gerados no Ubuntu, suspeitas do fenômeno de envelhecimento de software no ambiente do servidor com o SGBD PostgreSQL.

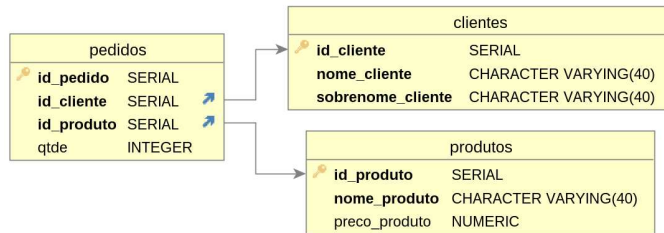


Fig. 1. Schema de banco de dados adotado no experimento.

Para o envio de requisições ao banco de dados de maneira automatizada e constante, foi utilizado o Apache Jmeter versão 5.4.1 [18]. O Jmeter é um software de código aberto muito utilizado em testes de estresse e para medir o desempenho de diversos ambientes computacionais. Para coletar os dados de uso de memória RAM e da CPU, foram desenvolvidos dois *Shell scripts* que a cada cinco segundos faziam a captura de tais informações. O comando *free* do Linux foi utilizado no *script* da coleta dos dados de memória RAM e o comando *mpstat* no *script* de coleta de dados da CPU. Para a coleta dos processos no ambiente, foi desenvolvido um *script* em Python para capturar a cada uma hora todos os processos executados no Ubuntu. Nesse *script*, foi utilizado o comando *smem*. O *smem* é um comando muito útil para monitorar o uso da memória RAM pelos processos do sistema operacional, oferecendo dados como USS (tamanho do conjunto único), PSS (tamanho do conjunto proporcional) e RSS (tamanho do conjunto residente).

C. Análises

Neste trabalho, foi realizada uma análise de tendência nos dados de uso de memória RAM e CPU. Com o auxílio da linguagem R, utilizando o pacote "trend", foi executado o teste de Mann-Kendall. Esse teste sequencial e não paramétrico é capaz de determinar se um conjunto de dados tem a tendência de crescimento ou decréscimo ao longo do tempo. Nesse caso, especificamente, a intenção foi verificar se houve degradação da memória no ambiente por conta do envio de requisições constantes de leitura ao PostgreSQL, visto que o crescimento no consumo de memória é um forte indicador de envelhecimento de software [5].

Para o teste de Mann-Kendall temos duas hipóteses: a hipótese nula, quando não existe tendência nos dados e a hipótese alternativa, quando os dados tem uma tendência de aumento ou de diminuição. O valor de p (*p-value*) é o que determina se aceitamos ou não a hipótese alternativa, que indica se existe ou não uma tendência nos dados. Caso o valor de *p-value* for inferior a 0,05, aceitamos a hipótese alternativa de que existe uma tendência de crescimento ou diminuição nos

dados. Já o valor de S determina se os dados tem a tendência de aumento ou diminuição. Quando o valor de S é positivo, existe uma tendência de aumento e se o valor de S for negativo, temos uma tendência de diminuição nos dados.

IV. RESULTADOS E ANÁLISES

Nesta seção, são apresentados os resultados obtidos com a execução dos experimentos e as análises realizadas para detectar indícios de envelhecimento de software.

A. Dados Básicos obtidos nos Testes

Para cada carga de trabalho aplicada, foram coletados dados básicos para uma melhor compreensão do trabalho. A Tabela III detalha alguns dos resultados coletados para as cargas de trabalho utilizadas. A tabela mostra o tempo de resposta médio das requisições em milissegundos, a vazão em requisições por segundos e a taxa de erro de cada teste, onde 100% das requisições foram atendidas pelo servidor, ou seja, 0% de erro.

TABELA III
DADOS OBTIDOS NOS TESTES.

Carga	Tempo de Resposta Médio	Vazão	Taxa de Erro
Baixa	31 ms	4,8 req/s	0%
Média	680 ms	42,3 req/s	0%
Alta	1932 ms	46,9 req/s	0%

B. Análise do Uso da Memória RAM

Como parte das análises para investigar se seria encontrado ou não indícios de envelhecimento de software no ambiente, foi monitorado por meio de um *script*, o uso da memória RAM no decorrer de cada teste. O *script* coletou a cada 5 segundos por meio do comando *free*, os dados de uso da memória. Sendo assim, foi possível com esses dados, entender o comportamento da memória RAM e realizar o teste de Mann-Kendall para detectar alguma degradação de memória por efeito das cargas no ambiente. Vale ressaltar que em nenhuma das cargas de trabalho que foram aplicadas ao servidor, o sistema operacional chegou a utilizar mais do que a memória total disponível, deste modo, a memória SWAP alocada não chegou a ser utilizada em nenhum momento, não sendo necessário considerar esse indicador de envelhecimento de software.

A Figura 2 apresenta o monitoramento da memória com a carga baixa, onde é possível observar como o uso da memória foi crescendo mesmo com a carga constante durante todas as 48 horas de envio de requisições. O percentual máximo de memória usada chegou próximo dos 13,5% já perto do fim da execução do experimento.

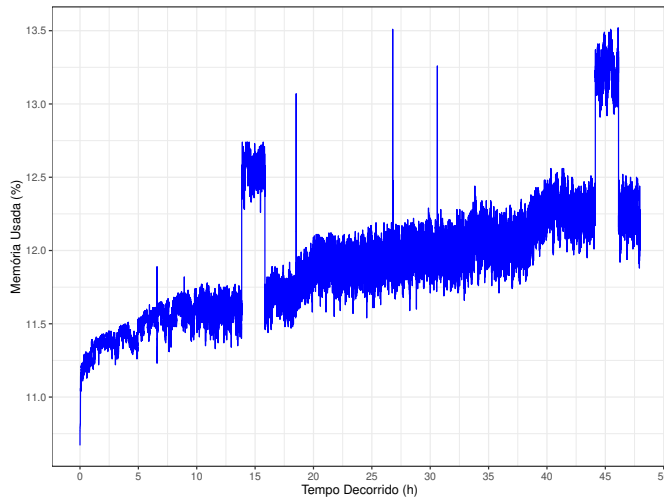


Fig. 2. Consumo de memória RAM no servidor - Carga Baixa.

Já no experimento com a carga média, onde tivemos um maior envio de requisições por segundo ao servidor, podemos ver por meio da Figura 3 um maior patamar de utilização da memória RAM, se comparado ao gráfico da Figura 2. Nesse gráfico, também é possível observar que o uso da memória foi se elevando com o passar do tempo de teste, atingindo alguns picos superiores a 15% de utilização da memória RAM.

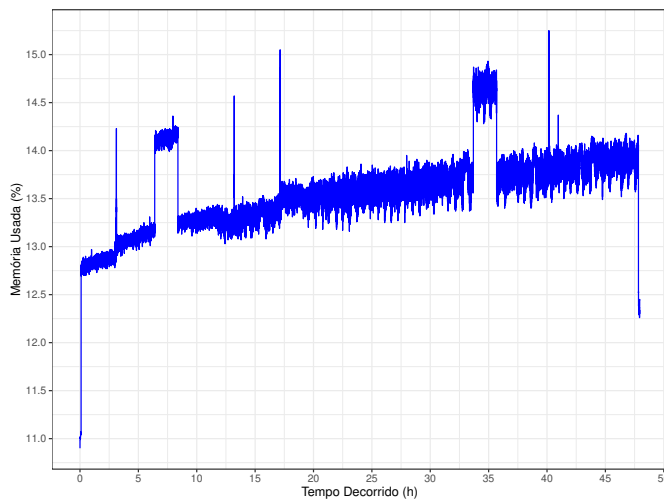


Fig. 3. Consumo de memória RAM no servidor - Carga Média.

Como nos outros gráficos anteriores com as cargas baixa e média, também é possível perceber no gráfico da Figura 4 um crescimento no percentual de uso de memória RAM ao longo do tempo em que se executou tal carga. Um maior patamar e maiores picos de uso de memória ocorreram quando é empregada uma carga de trabalho mais elevada ao banco de dados instalado no servidor. Também é possível observar por meio do gráfico, que ao iniciado o experimento, tivemos uma elevação abrupta no percentual de memória usada, partindo aproximadamente de 10,5% para acima de 14% de utilização da memória.

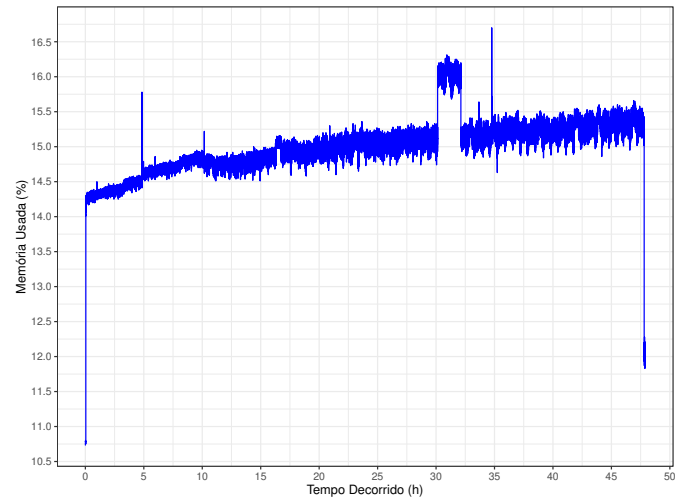


Fig. 4. Consumo de memória RAM no servidor - Carga Alta.

Para confirmar a suspeita de tendência de crescimento no uso da memória RAM ao longo do tempo nos gráficos das Figuras 2, 3 e 4, por ocasião das cargas de trabalho empregadas, foi realizado o teste de Mann-Kendall. A Tabela IV apresenta os resultados do teste de Mann-Kendall que foram obtidos com as cargas de trabalho empregadas ao servidor. Para todas as cargas, o valor de *p-value* foi inferior a 0,05, o que indica que temos nos dados alguma tendência, seja de crescimento ou de decréscimo. Como o valor de *S* foi positivo em todas as situações, é possível chegar à conclusão de que houve uma tendência de aumento ao longo do tempo no consumo de memória para todas as cargas de trabalho aplicadas ao servidor. Deste modo, é possível concluir que houve no ambiente, uma tendência de degradação de memória, gerando um acúmulo no decorrer do tempo em que o experimento foi aplicado.

TABELA IV
TESTE DE MANN-KENDALL PARA AS CARGAS DE
TRABALHO - MEMÓRIA RAM.

Carga	<i>p-value</i>	<i>S</i>	Tendência
Baixa	2,2e-16	3,88e+08	Crescimento
Média	2,2e-16	3,73e+08	Crescimento
Alta	2,2e-16	4,16e+08	Crescimento

C. Análise do Uso da CPU

Também foi monitorado o uso da CPU durante a execução do experimento com as três cargas de trabalho, com a finalidade de encontrar alguma tendência de degradação. O *script* utilizado para essa tarefa coletou a cada 5 segundos por meio do comando *mpstat* dados de uso da CPU no servidor de banco dados. O objetivo foi verificar por meio do teste de Mann-Kendall se existe durante a execução das cargas de trabalho uma tendência de crescimento no uso da CPU, o que seria mais um forte indício de sintomas de envelhecimento de software no ambiente.

A Figura 5 mostra o uso da CPU para a carga baixa. É possível notar um leve crescimento no uso do processador, porém baixo, ficando entre 5 e 10% na maior parte do tempo, com a presença de picos de uso superiores a 30%.

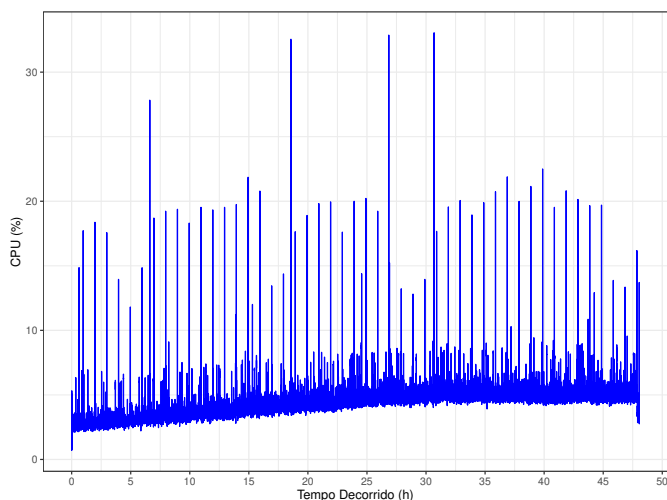


Fig. 5. Uso da CPU no servidor - Carga Baixa.

Já na Figura 6, notamos que o uso da CPU se estabeleceu na faixa dos 15% de uso com picos próximos aos 50%. Também é possível visualizar que de maneira discreta o uso da CPU foi se elevando no decorrer do tempo, porém um teste estatístico é necessário para confirmar a suspeita.

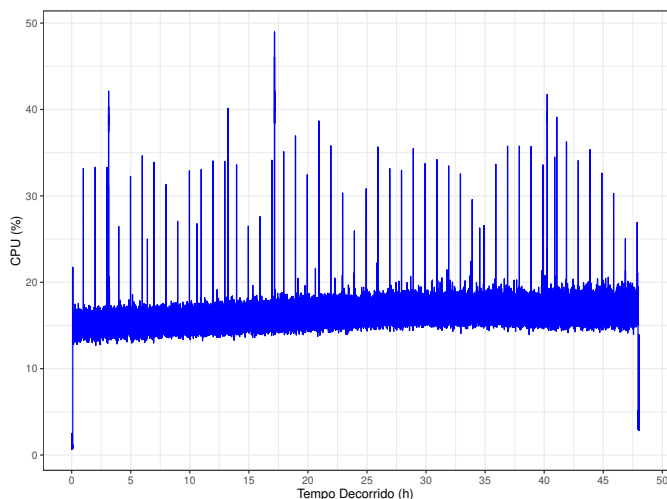


Fig. 6. Uso da CPU no servidor - Carga Média.

A Figura 7 apresenta o comportamento da CPU durante o teste com a carga alta. Mesmo com uma demanda de requisições mais elevada, percebemos que o uso do processador não eleva significativamente. É visível que durante a maior parte do experimento, somente 20% da CPU foi utilizada, tendo um único pico superior aos 50%. A tendência de crescimento é pouco perceptível nesse gráfico, reforçando a necessidade de um teste estatístico para confirmar tal suspeita.

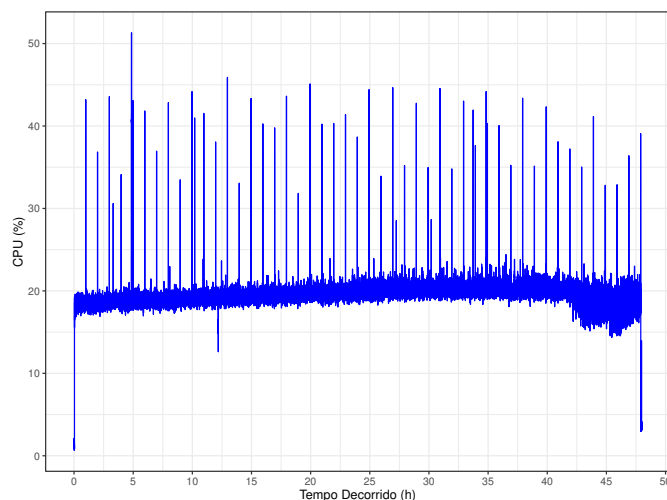


Fig. 7. Uso da CPU no servidor - Carga Alta.

Para confirmar a suspeita de tendência de crescimento no uso da CPU, também foi realizado o teste de Mann-Kendall para esses dados. A Tabela V mostra os resultados do teste estatístico de Mann-Kendall e confirma que o uso da CPU se elevou no decorrer dos experimentos com todas as cargas aplicadas. Tivemos o valor de *p-value* inferior a 0,05 e o valor de *S* positivo em todas as situações, o que indica que também houve um forte indício de envelhecimento de software no ambiente quando foi analisado uso da CPU no servidor de banco de dados com o PostgreSQL.

TABELA V
TESTE DE MANN-KENDALL PARA AS CARGAS DE TRABALHO - CPU.

Carga	<i>p-value</i>	<i>S</i>	Tendência
Baixa	2,2e-16	3,96e+08	Crescimento
Média	2,2e-16	1,69e+08	Crescimento
Alta	2,2e-16	1,71e+08	Crescimento

D. Análise dos Processos

Com o uso do *script* de monitoramento, foi coletado ao longo da execução de cada teste, os processos que estavam sendo executados no ambiente de banco de dados. Com esses dados, foi possível verificar quais os processos estavam consumindo mais memória RAM do servidor e saber se as requisições enviadas ao PostgreSQL estavam fazendo com que SGBD fosse o principal responsável pelo uso da memória. Para essa análise, foi utilizado o último arquivo coletado pelo *script* de monitoramento de processos, pois nele podemos encontrar um estágio em que o ambiente já está degradado pela alta quantidade de requisições enviadas pelo cliente, próximo às 48 horas de teste para uma dada carga.

É importante destacar que para classificar os processos de modo a gerar os cinco processos que mais consumiram memória com os dados do monitoramento de cada carga de trabalho, foi feito um somatório dos valores de RSS (tamanho do conjunto residente) de todos os processos relacionados a execução do PostgreSQL. Uma vez que somente pela execução

do SGBD no Ubuntu, temos sete processos rodando no sistema operacional, mesmo sem nenhum usuário conectado. Além disso, para cada usuário conectado ao SGBD, um processo é criado no Ubuntu. Por exemplo, se tivermos cem usuários conectados fazendo requisições ao SGBD, iremos encontrar no arquivo de monitoramento dos processos, além dos sete processos que são padrão do sistema, mais cem referentes a usuários conectados. Deste modo, o PID 0, denominado de "postgres" nas Tabelas VI, VII, VIII e nas Figuras 8, 9, 10, representa de forma única os processos do PostgreSQL já com os valores de RSS somados. Isso foi necessário para diferenciá-lo dos demais processos do Ubuntu, listados no arquivo coletado.

Na Tabela VI temos os cinco processos que mais consumiram memória no ambiente do servidor de banco de dados durante o teste com a carga baixa e a Figura 8 mostra a quantidade de memória usada por esses processos. Mesmo na segunda posição, o uso do PostgreSQL se destacou, pois chegou muito próximo de atingir o valor em megabyte de memória utilizada do processo *gnome-shell*. Outros processos tiveram um uso bem mais discreto do que os dois primeiros, sendo eles o *Xorg vtr1 -dis*, o *tracker-store* e o *systemd-journald*.

TABELA VI
OS CINCO PROCESSOS QUE MAIS CONSUMIRAM MEMÓRIA - CARGA BAIXA.

PID	Descrição
1125	gnome-shell - gráfico do ambiente da área de trabalho GNOME.
0	postgres - Soma dos Processos do PostgreSQL.
1060	Xorg vtr1 -dis - processo relacionado ao sistema de janelas X do Linux.
1592	tracker-store - processo de indexação de arquivos.
282	systemd-journald - processo que coleta e armazena dados de log.

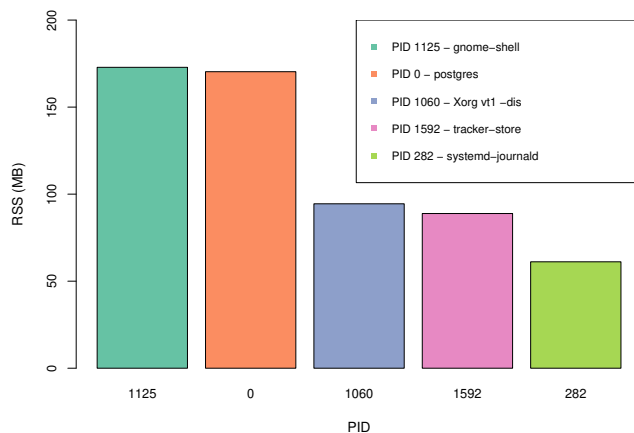


Fig. 8. Quantidade de memória utilizada pelos cinco processos que mais consumiram memória - Carga Baixa.

A Tabela VII apresenta os processos que mais consumiram memória com a carga de trabalho média. O uso da memória pelos processos do PostgreSQL se destacaram na quantidade de memória usada (ver Figura 9), chegando a consumir quase

1000 MB da memória RAM. Também podemos observar que o PostgreSQL consumiu bem mais da memória do que os demais processos.

TABELA VII
OS CINCO PROCESSOS QUE MAIS CONSUMIRAM MEMÓRIA - CARGA MÉDIA.

PID	Descrição
0	postgres - Soma dos Processos do PostgreSQL.
1134	gnome-shell - gráfico do ambiente da área de trabalho GNOME.
1072	Xorg vtr1 -dis - processo relacionado ao sistema de janelas X do Linux.
2171	tracker-store - processo de indexação de arquivos.
280	systemd-journald - processo que coleta e armazena dados de log.

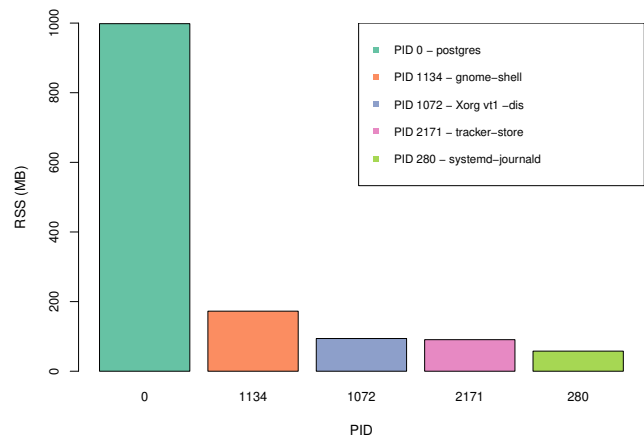


Fig. 9. Quantidade de memória utilizada pelos cinco processos que mais consumiram memória - Carga Média.

A Tabela VIII lista os processos que mais consumiram memória com a carga alta, enquanto a Figura 10 apresenta o consumo de tais processos. Como é visto na Figura 10, o valor de RSS do PID 0 se elevou ainda mais devido a quantidade de requisições enviadas ao servidor de banco de dados, atingindo quase 2000 MB da memória. Pela descrição da Tabela VIII é possível perceber que o PostgreSQL foi o principal responsável pelo uso da memória RAM do servidor.

TABELA VIII
OS CINCO PROCESSOS QUE MAIS CONSUMIRAM MEMÓRIA - CARGA ALTA.

PID	Descrição
0	postgres - Soma dos Processos do PostgreSQL.
1127	gnome-shell - gráfico do ambiente da área de trabalho GNOME.
1063	Xorg vtr1 -dis - processo relacionado ao sistema de janelas X do Linux.
1570	tracker-store - processo de indexação de arquivos.
281	systemd-journald - processo que coleta e armazena dados de log.

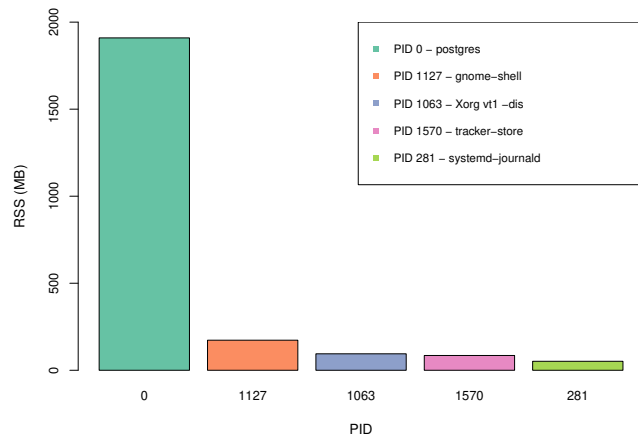


Fig. 10. Quantidade de memória utilizada pelos cinco processos que mais consumiram memória - Carga Alta.

Como podemos ver nas análises, as cargas de trabalho empregadas fizeram com que PostgreSQL fosse um dos principais responsáveis pelo uso da memória RAM, superando bastante os outros processos em carga média e alta. Deste modo, é possível concluir que as cargas de trabalho empregadas ao PostgreSQL, fizeram com que ele consumisse a maior parcela do uso da memória no servidor na maioria dos casos, o que indica que ele possa ser um dos principais responsáveis pelos efeitos de envelhecimento de software no ambiente.

E. 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. No entanto, acredita-se que o ambiente adotado é bastante semelhante aos ambientes adotados por empresas de pequeno e médio porte. Além disso, o procedimento de análise mostrado neste artigo pode ser facilmente adaptado a qualquer outro ambiente de banco de dados e pesquisadores com interesse podem replicar nosso estudo com o mesmo procedimento.

Limitação da duração e do número de execuções experimentais. Experimentos de envelhecimento de software são uma atividade demorada e sua duração é inevitavelmente limitada como em qualquer trabalho experimental. Embora tenha sido realizado testes com duração de 48 horas para cada carga de trabalho da Tabela II, não podemos garantir que todas as tendências de degradação sejam detectadas.

Representatividade da carga de trabalho adotada. A carga de trabalho adotada faz o uso do esquema de banco de dados apresentado na Figura 1. Assim, é possível que o leitor obtenha resultados diferentes dos nossos, se for considerado um esquema ou uma consulta diferente. Além disso, as cargas de trabalho consideradas em nosso estudo consistem em três níveis de intensidades constantes (baixo, médio e alto) que podem não representar comportamentos reais dos usuários. Como o objetivo deste trabalho é expor potenciais tendências de envelhecimento, não consideramos cargas de trabalho variáveis nos experimentos.

V. CONCLUSÃO

Neste trabalho, foi investigada a presença do fenômeno de envelhecimento de software em um ambiente de banco de dados, onde o PostgreSQL foi utilizado como SGBD. A utilização das cargas de trabalho empregadas ao servidor, potencializaram a degradação da memória, fazendo com que em 48 horas de execução para cada carga de trabalho constante, fosse mostrado um crescimento no consumo de memória RAM e CPU ao longo do tempo. O teste de Mann-Kendall foi realizado para confirmar estatisticamente a suspeita de crescimento no uso de recursos de hardware. Além disso, foi certificado por meio da análise dos processos do sistema operacional, que o PostgreSQL foi um dos principais responsáveis pelo consumo de memória RAM no decorrer dos experimentos.

Os resultados obtidos neste trabalho, mostram que existe um forte indicativo de suspeita de envelhecimento de software no ambiente do servidor de banco de dados e que essa degradação pode ser em decorrência da alta demanda enviada ao PostgreSQL. Existe a necessidade que investigações mais aprofundadas sejam feitas para que outros fatores sejam avaliados para corroborar com o estudo, tais como, consumo de energia e tempo de resposta. Para trabalhos futuros, um tempo maior de execução das cargas pode ser considerado, além do estudo de ambientes com outros SGBDs ou sistemas operacionais diferentes.

REFERÊNCIAS

- [1] DB-Engines, "DB-Engines Ranking," <https://db-engines.com/en/ranking>, 2022, [Online; accessed 11-nov-2022].
- [2] U. F. Minhas, S. Rajagopalan, B. Cully, A. Aboulnaga, K. Salem, and A. Warfield, "Remusdb: Transparent high availability for database systems," *The VLDB Journal*, vol. 22, no. 1, pp. 29–45, 2013.
- [3] H. He, "Tuning backfired? not (always) your fault: Understanding and detecting configuration-related performance bugs," in *Proceedings of the 2019 27th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, 2019, pp. 1229–1231.
- [4] D. L. Parnas, "Software aging," in *Proceedings of 16th International Conference on Software Engineering*. IEEE, 1994, pp. 279–287.
- [5] M. Grottko, R. Matias, and K. S. Trivedi, "The fundamentals of software aging," in *2008 IEEE International conference on software reliability engineering workshops (ISSRE Wksp)*. Ieee, 2008, pp. 1–6.
- [6] A. Bovenzi, D. Cotroneo, R. Pietrantuono, and S. Russo, "On the aging effects due to concurrency bugs: A case study on mysql," in *2012 IEEE 23rd International Symposium on Software Reliability Engineering*. IEEE, 2012, pp. 211–220.
- [7] H. B. Mann, "Nonparametric tests against trend," *Econometrica: Journal of the econometric society*, pp. 245–259, 1945.
- [8] M. G. Kendall, "Rank correlation methods." 1948.
- [9] E. Andrade, F. Machida, R. Pietrantuono, and D. Cotroneo, "Memory degradation analysis in private and public cloud environments," in *2021 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW)*. IEEE, 2021, pp. 33–39.
- [10] —, "Software aging in image classification systems on cloud and edge," in *2020 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW)*. IEEE, 2020, pp. 342–348.
- [11] D. Dias and E. Andrade, "Análise de envelhecimento de software em uma plataforma de blockchain," in *Anais do V Workshop em Blockchain: Teoria, Tecnologias e Aplicações*. SBC, 2022, pp. 40–53.
- [12] C. Melo, F. Oliveira, J. Dantas, J. Araujo, P. Pereira, R. Maciel, and P. Maciel, "Performance and availability evaluation of the blockchain platform hyperledger fabric," *The Journal of Supercomputing*, pp. 1–23, 2022.
- [13] L. Vinícius, L. Rodrigues, M. Torquato, and F. A. Silva, "Docker platform aging: a systematic performance evaluation and prediction of resource consumption," *The Journal of Supercomputing*, pp. 1–31, 2022.

- [14] F. Oliveira, J. Araujo, R. Matos, and P. Maciel, "Software aging in container-based virtualization: an experimental analysis on docker platform," in *2021 16th Iberian Conference on Information Systems and Technologies (CISTI)*. IEEE, 2021, pp. 1–7.
- [15] S. Huo, D. Zhao, X. Liu, J. Xiang, Y. Zhong, and H. Yu, "Using machine learning for software aging detection in android system," in *2018 Tenth International Conference on Advanced Computational Intelligence (ICACI)*. IEEE, 2018, pp. 741–746.
- [16] Y. Qiao, Z. Zheng, and F. Qin, "An empirical study of software aging manifestations in android," in *2016 IEEE international symposium on software reliability engineering workshops (ISSREW)*. IEEE, 2016, pp. 84–90.
- [17] H. Couto, F. Silva, G. Callou, and E. Andrade, "Uma abordagem experimental para avaliar o desempenho do banco de dados open-source postgresql," in *Anais da X Escola Regional de Informática de Goiás*. SBC, 2022, pp. 12–23.
- [18] A. JMeter™, "Apache JMeter - What can I do with it?" <https://jmeter.apache.org/>, 2022, [Online; accessed 11-nov-2022].



Herderson Couto is a technology technician at the Federal University of Pernambuco, Brazil. In 2019, he finished his specialization in Information Security. His research interest is focused in Database, Performance, and Software Aging.



Ermeson Andrade is an associate professor at the Department of Computing at the Federal Rural University of Pernambuco, Brazil. In 2014, he completed his PhD in Computer Science from the Federal University of Pernambuco. His research interest is focused in Performance, Dependability, and Modeling.



Francisco Airton Silva is a professor at the Federal University of Piauí (campus Picos), Brazil. In 2017, he completed his PhD in Computer Science from the Federal University of Pernambuco. His research interest is focused in Cloud Computing, Mobile Computing, and Performance Evaluation.



Gustavo Callou is an associate professor at the Department of Computing at the Federal Rural University of Pernambuco, Brazil. In 2013, he completed his PhD in Computer Science from the Federal University of Pernambuco. His research interest is focused in Performance and Dependency Modeling, Cloud Computing, Data Centers and Sustainability.