

Comparação de Serviços de Banco de Dados em Nuvem com RSSF: Amazon RDS e Google Cloud SQL

Marcelo Alexandre da Cruz Ismael
Universidade Tecnológica Federal do
Paraná (UTFPR)
Toledo, PR, Brasil
marceloismael@utfpr.edu.br

Edson Tavares de Carmargo
Universidade Tecnológica Federal do
Paraná (UTFPR)
Toledo, PR, Brasil
edson@utfpr.edu.br

Guilherme Galante
Universidade Estadual do Oeste do
Paraná (UNIOESTE)
Cascavel, PR, Brasil
guilherme.galante@unioeste.br

César Alberto da Silva
Instituto Federal de Educação, Ciência
e Tecnologia (IFSP)
Presidente Epitácio, SP, Brasil
cesar@ifsp.edu.br

Fabrcio Fernando Alves
Instituto Federal de Educação, Ciência
e Tecnologia (IFSP)
Presidente Epitácio, SP, Brasil
fabricio.alves@ifsp.edu.br

Luis Carlos Erpen de Bona
Universidade Federal do Paraná
(UFPR)
Curitiba, PR, Brasil
bona@inf.ufpr.br

Abstract— Cloud computing has brought the possibility of having computing resources as a service, unlike how we are used to, where they are provided as a product. One of the services provided by cloud providers is relational database management, which is an indispensable component in most computing systems nowadays. As there are several providers that offer the same service, then there is a need for a comparison between them, in order to show which platform is feasible according to the need of the application. This paper details two cloud relational database services, Amazon RDS and Google Cloud SQL, showing how they works, what their limitations are, and how the costing of both is performed. With the understanding of the two services a comparison is made between them according to pre-defined metrics. The Amazon RDS service performed better in the execution time of the instructions against the Google Cloud SQL service. Companies that need to choose the best relational database service can use this search as an indicator.

Keywords— cloud computing; database, wireless sensor network; amazon RDS; google cloud SQL.

Resumo— A computação em nuvem trouxe a possibilidade de ter recursos computacionais como um serviço, diferentemente do que estamos acostumados, onde eles são fornecidos como um produto. Um dos serviços oferecidos pelos provedores de nuvem é a gestão de banco de dados relacional, que é um componente indispensável na maioria dos sistemas computacionais hoje em dia. Como existem vários provedores que oferecem o mesmo serviço, surge a necessidade de uma comparação entre eles, a fim de mostrar qual plataforma é viável de acordo com a necessidade da aplicação. Este artigo detalha dois serviços de banco de dados relacional em nuvem, Amazon RDS e Google Cloud SQL, mostrando como eles funcionam, quais são suas limitações e como o custo de ambos é calculado. Com a compreensão desses dois serviços, é feita uma comparação entre eles de acordo com métricas pré-definidas. O serviço Amazon RDS teve um melhor desempenho no tempo de execução das instruções em comparação ao serviço Google Cloud SQL. Em-

presas que precisam escolher o melhor serviço de banco de dados relacional podem usar esta pesquisa como um indicador.

Palavras Chaves— computação em nuvem; base de dados; Redes sensores sem fio; Amazon RDS, Google Cloud SQL

I. INTRODUÇÃO

O uso de tecnologias de Rede de Sensores Sem Fios (RSSF) para sensoriamento ubíquo permite medir, inferir e compreender um determinado ambiente, seja em uma área urbana ou ecológica [1][2]. Os nós sensores são equipados com sensores que permitem o monitoramento de diversas características de um ambiente, como por exemplo, som, imagem, temperatura, pressão, luminosidade, níveis de ruído, umidade do solo, poluentes químicos, etc.[2]. Seu uso permite gerar uma enorme quantidade de dados que devem ser armazenados para posteriormente serem processados e apresentados de modo interpretável [2].

Uma maneira de armazenar esses dados é com a utilização dos serviços disponibilizados pela computação em nuvem, assim como o serviço de banco de dados relacional [3] [4] [5].

Nesse cenário surgiram vários provedores com diferentes plataformas que fornecem esse mesmo tipo de serviço em nuvem [6][7]. Como consequência, a falta de evidências empírica faz com que a escolha do serviço seja feita de forma ad hoc. Esse artigo preenche essa lacuna, investigando o seguinte tema de pesquisa: “O serviço de Banco de Dados Relacional em Nuvem utilizando o mesmo SGBD entre provedores diferentes são equivalentes em relação a desempenho?”. Para responder essa questão foi realizado um experimento que compara o desempenho dos serviços de Banco de Dados Relacional em nuvem Amazon RDS¹ e Google Cloud SQL² utilizando MySql como SGBD.

¹<https://aws.amazon.com/pt/rds>

²<https://cloud.google.com/products/databases>

Esse artigo apresenta uma avaliação de dois dos serviços de Banco de Dados Relacional em Nuvem amplamente utilizado na indústria e na academia [8] [9]. Assim tendo como contribuição: (i) avaliação empírica do desempenho dos serviços de banco de dados relacionais em nuvem, (ii) implementação de uma arquitetura em nuvem para Rede de Sensores sem Fios (RSSF).

As próximas seções deste artigo estão organizadas da seguinte maneira: na Seção II são abordados os trabalhos relacionados. Na Seção III são apresentados o planejamento e a operação do experimento. Os resultados do experimento são apresentados na Seção IV. Por fim, na Seção V são apresentadas a conclusão e os trabalhos futuros.

II. TRABALHOS RELACIONADOS

Muitas pesquisas foram feitas para analisar os diferentes serviços em nuvem [10] [11] assim como solução para investigar a portabilidade [12] e adoção de múltiplas nuvens [12]. No entanto, pouca pesquisa foi feita para fornecer dados concretos do desempenho do serviço de banco de dados relacional em nuvem.

Nesta seção, apresentamos brevemente três estudos relacionados a esse artigo, destacando suas diferenças. Ao realizar um experimento [13], examina três principais plataformas de banco de dados em nuvem—Microsoft Azure³, IBM DB2⁴ e Oracle Cloud⁵ focando em suas características, desempenho e adequação a diversos casos de uso.

O trabalho desenvolvido em [14] investiga o desempenho de bancos de dados relacionais (MySQL e PostgreSQL) em comparação com bancos de dados de chave-valor (Memcached e Redis) sob diferentes cargas de teste. Utilizando o Yahoo! Cloud Serving Benchmark. Outro estudo apresentado em [15] discute a avaliação de desempenho de sistemas de banco de dados distribuídos em ambientes de computação em nuvem. Desta forma apresenta os desafios enfrentados, como latência de rede e complexidade de acesso a dados, ressaltando a importância de uma avaliação adequada para garantir a eficiência dos sistemas.

Por fim, este artigo de maneira empírica analisa os serviços de banco de dados relacionais em nuvem, comparando especificamente o Amazon RDS e o Google Cloud SQL. Ele investiga como esses serviços funcionam, suas limitações e custos, com foco no tempo de execução além de apresentar uma solução de banco de dados em nuvem integrada a uma RSSF. Além disso artigo justifica a necessidade de comparações entre plataformas, enfatizando que a escolha do serviço adequado pode impactar significativamente a eficiência operacional de uma empresa.

III. PLANEJAMENTO E OPERAÇÃO

O objetivo desse estudo é avaliar qual serviço de banco de dados relacional em nuvem possui o melhor desempenho.

³<https://azure.microsoft.com/>

⁴<https://www.ibm.com/db2>

⁵<https://www.oracle.com/br/cloud/>

Desta forma para atingir esse objetivo foi desenvolvido uma aplicação em Java que se comunica com os serviços de banco de dados: Amazon RDS e Google Cloud SQL. A finalidade dessa aplicação é executar consultas SQL para assim calcular o tempo gasto de execução para cada um dos serviços. Para mitigar aspectos como latência da rede, devido à distância geográfica entre aplicação local e os *Data Centers*, foi escolhido a região Sul dos EUA, pois ambos provedores Amazon e Google oferecem o serviço de banco de dados para mesma região.

Outro ponto relevante, que pode impactar no desempenho da rede, refere-se ao sinal *wi-fi*, que pode sofrer oscilações no momento do envio das consultas. Portanto o experimento foi executado dentro de um ambiente controlado utilizando uma rede cabeada. Esse experimento foi executado em 4 dias em horários distintos com objetivo de simular um ambiente real.

As configurações de hardware e software que suportaram a aplicação são apresentadas na Tabela I. A aplicação foi executada em um computador local e por meio da rede de internet se comunica com os serviços de banco de dados na nuvem. Para simular a RSSF foi utilizado um simulador denominado Castália⁶. Esse simulador é um framework que permite aos usuários testar e validar algoritmos, protocolos e aplicações para RSSF [16]. De acordo com os profissionais da área é o mais realista para RSSF. Seu módulo de rádio é baseado em rádios reais para dispositivos embarcados de baixa potência, tais como nós sensores sem fio [17].

TABELA I
CONFIGURAÇÃO DO COMPUTADOR UTILIZADO PARA EXECUÇÃO DO EXPERIMENTO

Item	Configurações
Processador	Intel® Core™ i5-5200U CPU 2.20 GHZ
Memória volátil	8 GB
Memória não volátil	500 GB SSD
Sistema operacional	Windows 11 - 64 bits
Simulador	Castalia 3.0

Para as configurações da aplicação, foi utilizado um banco de dados público⁷ com mais de cinco milhões de registros para executar as consultas SQL. A escolha desse banco de dados, permite uma análise mais precisa do desempenho de cada serviço de armazenamento em nuvem ao lidar com grandes volumes de dados. Tanto no Amazon RDS quanto no Google Cloud SQL, foi criada uma instância MySQL, garantindo que o ambiente e as características do banco de dados sejam consistentes em ambas as plataformas. Com isso, é possível fazer uma comparação justa entre os dois serviços em termos de tempo de resposta, eficiência e capacidade de lidar com cargas de trabalho intensivas. A Figura 1. ilustra essa configuração geral do experimento.

⁶<https://omnetpp.org/download-items/Castalia.html>

⁷<https://dev.mysql.com/doc/employee/en/employeesinstallation.html>

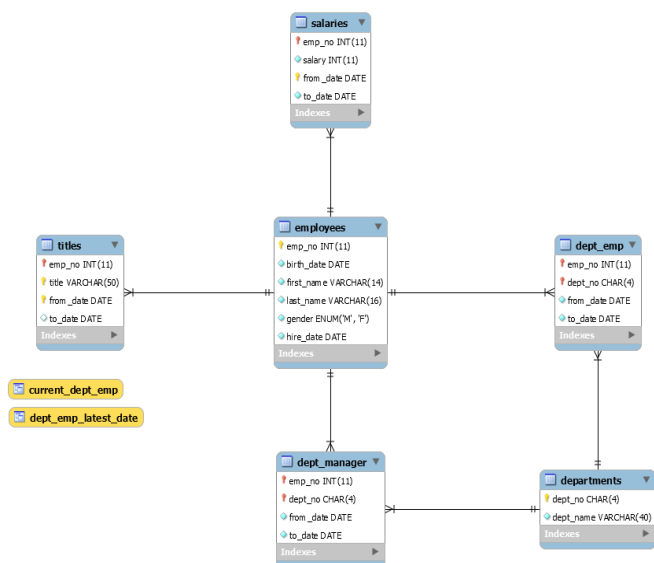


Fig. 1. Modelagem do bancos de dados employee.

As consultas foram criadas com a finalidade de simular diferentes situações e para analisar a eficiência de cada tipo de consulta nos diferentes serviços de banco de dados. Na Tabela II são representadas as tabelas do banco de dados que foram utilizadas nas consultas do experimento com a quantidade de registros.

TABELA II
LISTA DE TABELAS UTILIZADAS NO EXPERIMENTO

Tabelas	Registros
Departamentos	9
Dept_Emp	331603
Dept_Manager	24
Employees	300024
Salaries	2844047
Titles	443308

A Tabela III apresenta detalhadamente os seis tipos de instruções que foram definidos para a execução do experimento. Cada instrução foi projetada para simular diferentes operações que interagem com o banco de dados, como consultas de seleção, inserção, atualização e exclusão de registros.

TABELA III
LISTA DE INSTRUÇÕES

Instrução	Query
A	SELECT * FROM employees
B	SELECT * FROM salaries s INNER JOIN employees e ON s.emp_no = e.emp_no
C	SELECT * FROM salaries s INNER JOIN employees e ON s.emp_no = e.emp_no WHERE s.salary > 10000

D	DELETE FROM departments WHERE dept_no = "d010";
E	INSERT INTO departments VALUE ("d010", "Tecnologia da Informação")
F	UPDATE salaries SET salary = salary + 200 WHERE salary > 10000

Os resultados dessas consultas foram armazenados em um banco de dados local, contendo informações detalhadas sobre cada execução. Esses dados incluem o nome do data center onde a consulta foi processada, a data exata em que a consulta foi realizada, o tipo específico de query executada e o tempo de resposta medido em milissegundos. Isso permite uma análise precisa do desempenho de cada consulta em diferentes condições. A Fig. 2. oferece uma representação gráfica da topologia do experimento, ilustrando como as consultas foram distribuídas e processadas nos diversos data centers envolvidos no estudo.

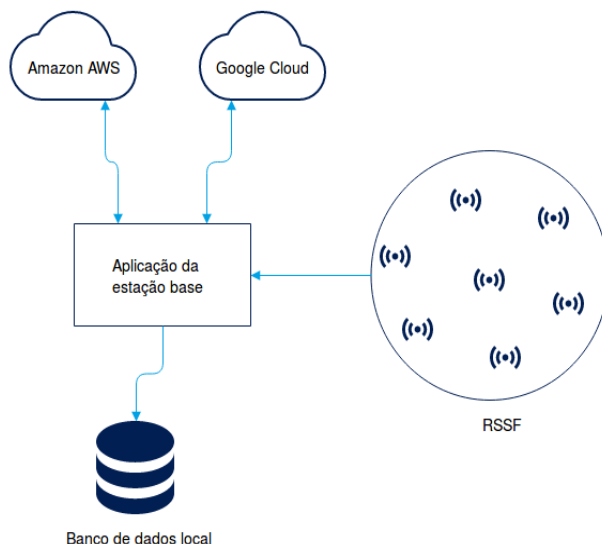


Fig. 2. Topologia do experimento.

A aplicação possui uma classe de conexão para cada serviço de banco de dados, que se conecta utilizando o driver JDBC do MySQL. A Fig. 3 apresenta parte do código responsável pelo envio das consultas. Uma estrutura de repetição (linha 2, Fig. 3), varia de acordo com a determinada consulta, conforme apresentada nas Figuras 4., 5. e 6. Não existe um consenso na literatura para definir a quantidade de repetições necessárias na execução de um experimento, portanto a quantidade de repetições para cada instrução foi definida de acordo com seu tempo de execução com objetivo que o experimento pudesse ser realizado no mesmo dia. As instruções A, B e C foram repetidas 30 vezes cada uma e as instruções D, E, e F foram repetidas 90 vezes.

A execução das instruções foi realizada de forma aleatória. As instruções são armazenadas aleatoriamente em um vetor e posteriormente o vetor é percorrido até todas as instruções terem sido executadas. A instrução selecionada é

passada por parâmetro para os dois serviços avaliados (linhas 4 e 5, Fig. 3) onde a conexão é realizada. Após os dois serviços conectados é feita a criação das *threads* (linha 8 até linha 13). Essas *threads* irão iniciar ao mesmo tempo e só serão reiniciadas novamente quando o último serviço terminar executar (linha 15 à linha 20, Figura 3). As Figuras 4., 5. e 6. ilustram a execução do experimento.

```

1 ...
2 for (int i=0; i<vetor.length; i++)
3 {
4     CountdownLatch startSignal = new CountdownLatch(1);
5     CountdownLatch doneSignal = new CountdownLatch(2);
6     queryEscolhida = query.randomQuery();
7     System.out.println ("\n"+ 1);
8
9     Thread t1 = new ConsultaAmazon (startSignal, doneSignal,
10     queryEscolhida, local, Amazon);
11
12     Thread t2 = new ConsultaGoogle (startSignal, doneSignal,
13     queryEscolhida, local, Google);
14
15     t1.start();
16
17     t2.start();
18
19     startSignal.coutDown();
20     doneSignal.await();
21
22 }
    
```

Fig. 3. Trecho de código do experimento.

Primeiramente, para as instruções classificadas como tipos A, B e C, conforme mostrado na Figura 4., são utilizadas diferentes consultas de seleção (como Select 1, Select 2 e Select 3) especificadas na Tabela III. A ordem de execução dessas consultas é definida aleatoriamente a cada ciclo, garantindo que a sequência em que cada uma é executada varie.

Uma vez determinada essa ordem aleatória, ambas as consultas são executadas simultaneamente nos dois bancos de dados em nuvem, de modo a comparar o desempenho e os resultados. Cada execução é registrada, e o processo se repete até que cada consulta de seleção tenha sido executada 30 vezes, gerando uma base de dados de resultados para análise.

Em relação às instruções de modificação, que incluem *Delete*, *Insert* e *Update* e estão detalhadas nas Figuras 5. e 6., a abordagem muda um pouco. Essas operações são executadas um total de 90 vezes para cada tipo de instrução.

Esses dados são essenciais para entender como cada plataforma lida com cargas de trabalho semelhantes e ajudam a fundamentar a escolha de um serviço de banco de dados em nuvem mais adequado para necessidades específicas.

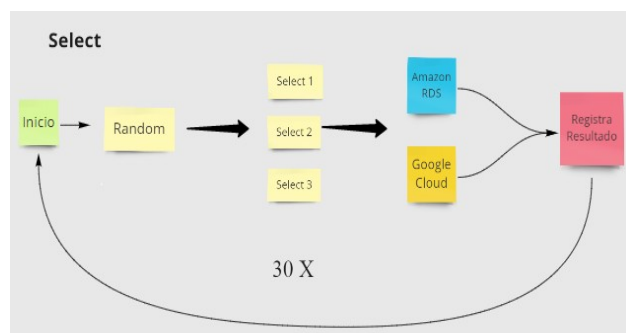


Fig. 4. Ordem de execução das instruções A, B, C

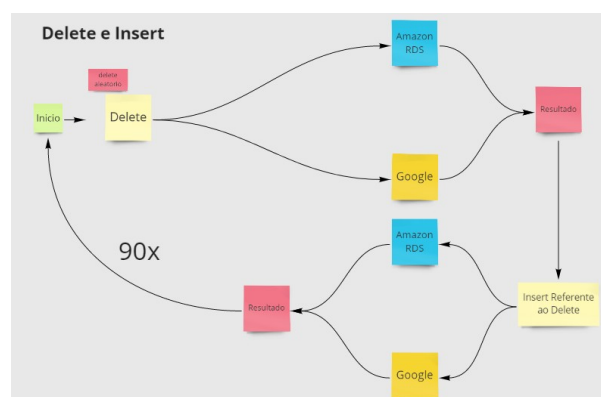


Fig. 5. Ordem de execução das instruções D e E.

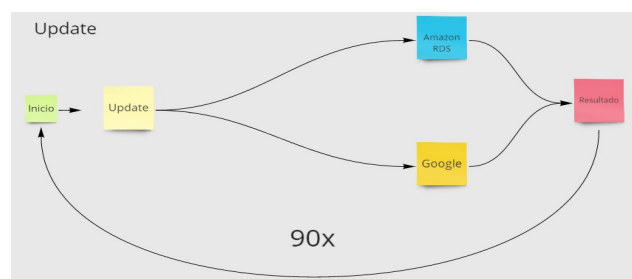


Fig. 6. Ordem de execução instrução F.

IV. RESULTADOS

A análise de desempenho dos serviços de banco de dados relacionais foi feita com o auxílio de diagramas de caixa (*boxplots*). Para geração dos gráficos foi utilizado o software de estatística computacional denominado R⁸.

Em relação à execução da instrução A, apresentadas na Fig. 7., nota-se que ambos os diagramas sugerem uma distribuição de dados positivamente assimétrica, com amplitude acentuadamente menor no tempo de execução por parte do banco de dados Amazon RDS.

⁸<https://www.r-project.org/>

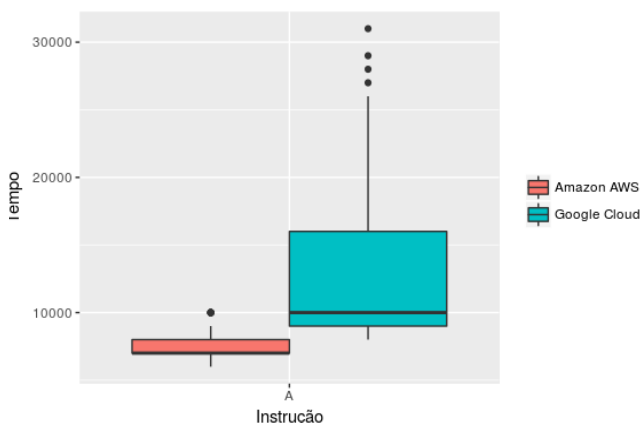


Fig. 7. Resumo da instrução A.

Nas instruções B e D, conforme ilustrado nas Figuras 8. e 9., não foram observadas diferenças significativas no desempenho entre os dois serviços de banco de dados em nuvem analisados. Isso indica que, para essas operações específicas de consulta e exclusão, ambos os serviços apresentaram tempos de resposta bastante similares, sem variações que afetassem a comparação. Já para a execução da instrução F, mostrada na Figura 10., o diagrama correspondente também sugere que os dois serviços tiveram um desempenho equivalente. Esse resultado reforça a ideia de que, em certas operações, ambos os serviços oferecem uma performance bastante homogênea.

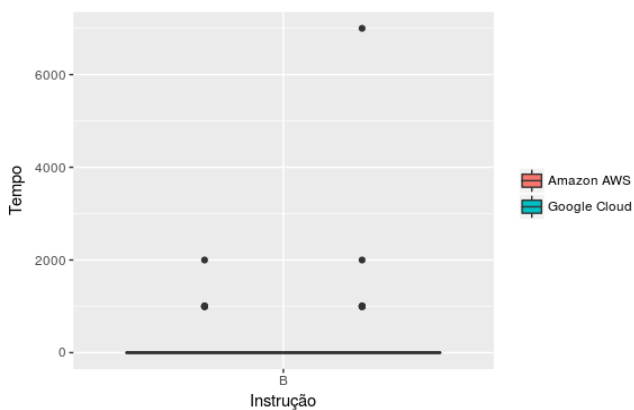


Fig. 8. Resultado da instrução B.

No que se refere à instrução C, conforme ilustrado na Fig. 11., o diagrama referente ao Amazon RDS indica uma distribuição de dados assimétrica negativa, o que significa que a maioria dos tempos de execução das consultas está concentrada em valores mais baixos, com alguns valores mais altos estendendo a cauda da distribuição. Em contraste, o diagrama do Google Cloud SQL apresenta uma distribuição assimétrica positiva, com a maioria dos tempos de execução concentrados em valores mais elevados e alguns tempos mais rápidos estendendo a cauda inferior. Além disso, observa-se que o Amazon RDS tem uma menor variação no tempo de execução das consultas, evidenciando maior consistência e um desempenho potencialmente mais

estável em relação ao Google Cloud SQL.

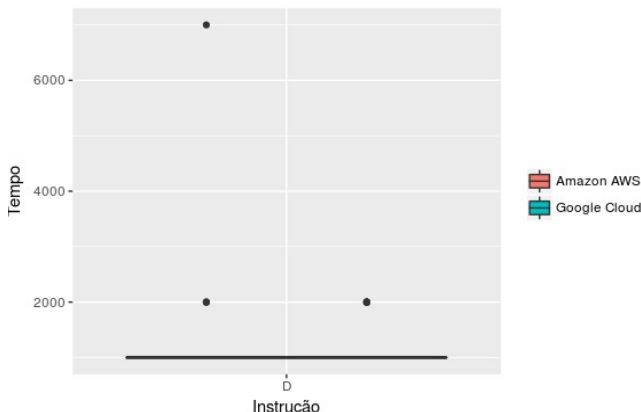


Fig. 9. Resultado da instrução D.

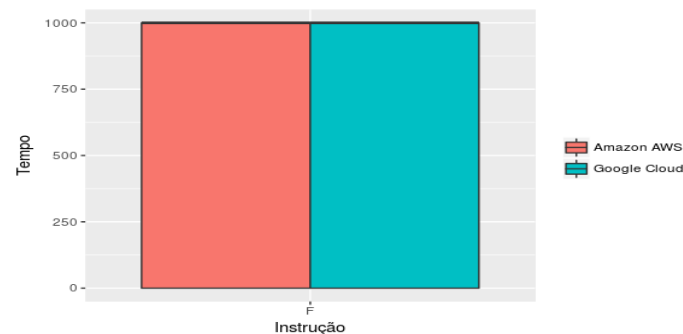


Fig. 10. Resultado da Instrução F.

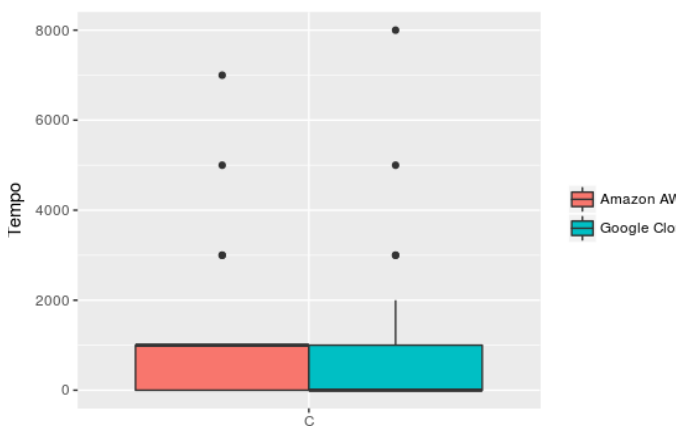


Fig. 11. Resultado da Instrução C.

Os diagramas da instrução E, conforme Fig. 12., também apontam para uma distribuição de dados positivamente assimétrica e, desconsiderando-se os valores discrepantes, o serviço do Google Cloud SQL para essa instrução possui uma menor amplitude no tempo de execução. No entanto, o tempo de execução do Amazon RDS para a maioria das amostras é menor. Além disso, há a ocorrência de valores discrepantes (*outliers*) nas instruções de A até E, o que se deve, possivelmente, à variação da banda de internet.

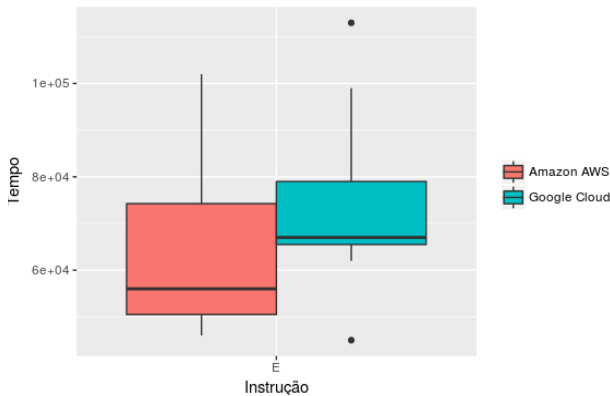


Fig. 12. Resultado da Instrução E.

V. CONCLUSÃO E TRABALHOS FUTUROS

Este artigo apresenta um experimento cujo objetivo é avaliar se os serviços de Banco de Dados Relacional em Nuvem, Amazon RDS e Google Cloud SQL, apresentam desempenho equivalente ao utilizar o mesmo Sistema de Gerenciamento de Banco de Dados (SGBD). Os resultados do experimento indicam que, mesmo após a exclusão de valores discrepantes (outliers), o Amazon RDS demonstrou um desempenho superior em termos de tempo de execução das instruções quando comparado ao Google Cloud SQL.

Para confirmar e generalizar esses resultados além de entender melhor as causas das diferenças observadas, propomos seguir três etapas adicionais. Primeiramente, o experimento será ampliado em três direções: (i) incluir a avaliação de outros provedores de nuvem, como Microsoft Azure, IBM e Rackspace; (ii) testar o desempenho de outros SGBDs; e (iii) avaliar o consumo de memória e processamento das instâncias no momento das consultas.

No contexto de implementação de uma arquitetura de Redes de Sensores Sem Fio (RSSF) utilizando bancos de dados em nuvem, foi possível verificar que ambos os serviços, Amazon RDS e Google Cloud SQL, permitem a integração. Isso abre a possibilidade de adicionar outros serviços típicos da computação em nuvem, como gerenciamento, elasticidade e redundância. Além disso, como trabalho futuro, sugere-se investigar o desempenho de bancos de dados relacionais em nuvem em comparação com bancos de dados não relacionais dentro de uma RSSF, para identificar potenciais diferenças e otimizações.

AGRADECIMENTOS

Gostaria de expressar meus sinceros agradecimentos à Universidade Federal do Paraná (UFPR), à Universidade Tecnológica Federal do Paraná (UTFPR) e à Universidade Estadual do Oeste do Paraná (Unioeste) pelo apoio e pelas oportunidades oferecidas ao longo da realização deste trabalho

REFERÊNCIAS

[1] S. Okdem and H. Shi, "Improving IoT and WSN Communication Throughput Using Evolutionary Optimization," 2024 6th International

Conference on Computer Communication and the Internet (ICCCI), Tokyo, Japan, 2024, pp. 169-174.

[2] A. Gupta and T. Kumar Dubey, "Performance Analysis of Clustering Protocols for WSN Assisted IoT Network," 2023 14th International Conference on Computing Communication and Networking Technologies (ICCCNT), Delhi, India, 2023, pp. 1-7

[3] B. Wibowo, A. Haq and M. T. A. A. Zein, "A Comparative Study of Load Balancing Methods in Cloud-Based Village Information Systems," 2023 IEEE 7th International Conference on Information Technology, Information Systems and Electrical Engineering (ICITISEE), Purwokerto, Indonesia, 2023, pp. 285-289.

[4] P. J. Assudani et al., "Testing Resiliency of AWS Services Using the Concept of Chaos Engineering," 2024 5th International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV), Tirunelveli, India, 2024, pp. 352-356, doi: 10.1109/ICICV62344.2024.00061.

[5] S. C. Muppalla, S. Rana and J. Chawla, "Cloud-Powered Blood Bank Management-Leveraging AWS Services for Efficiency and Scalability," 2023 3rd International Conference on Smart Generation Computing, Communication and Networking (SMART GENCON), Bangalore, India, 2023, pp. 1-6.

[6] A. Rajput, P. Gupta, P. Ghodeswar, S. Varma, K. K. Sharma and U. Singh, "Study of Cloud Providers (Azure, Amazon, and Oracle) According To Service Availability and Price," 2023 3rd International Conference on Pervasive Computing and Social Networking (ICPCSN), Salem, India, 2023, pp. 1177-1188.

[7] A. Choudhary, P. K. Verma and P. Rai, "Comparative Study of Various Cloud Service Providers: A Review," 2022 International Conference on Power, Energy, Control and Transmission Systems (ICPECTS), Chennai, India, 2022, pp. 1-8, doi: 10.1109/ICPECTS56089.2022.10047594.

[8] B. Wibowo, A. Haq and M. T. A. A. Zein, "A Comparative Study of Load Balancing Methods in Cloud-Based Village Information Systems," 2023 IEEE 7th International Conference on Information Technology, Information Systems and Electrical Engineering (ICITISEE), Purwokerto, Indonesia, 2023, pp. 285-289.

[9] S. C. Muppalla, S. Rana and J. Chawla, "Cloud-Powered Blood Bank Management-Leveraging AWS Services for Efficiency and Scalability," 2023 3rd International Conference on Smart Generation Computing, Communication and Networking (SMART GENCON), Bangalore, India, 2023,

[10] Z. Li, L. O'Brien, H. Zhang, and R. Cai, "On a Catalogue of Metrics for Evaluating Commercial Cloud Services," in 2012 ACM/IEEE 13th International Conference on Grid Computing, 2012, pp. 164-173.

[11] N. A. de Santana, F. A. A. Lins, and E. T. G. de Sousa, "Performance Evaluation of Mobile Applications in Mobile Cloud Environments," IEEE Lat. Am. Trans., vol. 14, no. 11, pp. 4597-4602, Nov. 2016.

[12] G. C. Silva, L. M. Rose, and R. Calinescu, "A Systematic Review of Cloud Lock-In Solutions," in 2013 IEEE 5th International Conference on Cloud Computing Technology and Science, 2013, vol. 2, pp. 363-368.

[13] Abiodun, Moses Kazeem. "A Comparative Study of Cloud Databases: Analyzing Microsoft Azure, IBM db2, and Oracle Cloud." Privacy Preservation and Secured Data Storage in Cloud Computing. IGI Global, 2023. 42-65

[14] D. Almeida et al., "Performance Comparison of Redis, Memcached, MySQL, and PostgreSQL: A Study on Key-Value and Relational

- Databases," 2023 Second International Conference On Smart Technologies For Smart Nation (SmartTechCon), Singapore, Singapore, 2023, pp. 902-907
- [15] N. Prasetio, D. Ferdinand, B. C. Wijaya, M. S. Anggreainy and A. Kumiawan, "Performance Analysis of Distributed Database System in Cloud Computing Environment," 2023 IEEE 9th International Conference on Computing, Engineering and Design (ICCED), Kuala Lumpur, Malaysia, 2023,
- [16] S. Verma, S. Kaur and A. D. Gupta, "A Brief Survey on Simulators for Designing Routing Protocol for Wireless Sensor Network," 2023 14th International Conference on Computing Communication and Networking Technologies (ICCCNT), Delhi, India, 2023.
- [17] S. Verma, S. Kaur and A. D. Gupta, "A Brief Survey on Simulators for Designing Routing Protocol for Wireless Sensor Network," 2023 14th International Conference on Computing Communication and Networking Technologies (ICCCNT), Delhi, India, 2023, pp. 1-6.