

# Análise de desempenho em serviços gratuitos na nuvem utilizando os provedores AWS e GCP

Karina Wiechork<sup>1</sup>, Darlan Eziquiel Felisberto da Silva<sup>1</sup>, Marco Antoni<sup>2</sup>

<sup>1</sup> Instituto Federal de Educação, Ciência e Tecnologia Farroupilha  
Frederico Westphalen, RS, Brasil

<sup>2</sup>Instituto Federal do Rio Grande do Sul  
Rolante, RS, Brasil

{karinawiechork, darlanfelisberto, marco.antoni910}@gmail.com

**Resumo.** *A computação em nuvem proporciona serviços sob demanda e pagamento de acordo com o uso. Analisar o desempenho desses provedores é um indicador importante que deve ser investigado antes de destinar a migração de um sistema para um ambiente em nuvem. Este artigo tem como objetivo analisar e apresentar o desempenho de IaaS, FaaS e PaaS, utilizando a opção gratuita, nos provedores de nuvem AWS e GCP. Com base nas configurações testadas, os resultados mostram que o provedor da Amazon teve um desempenho melhor em comparação com o Google.*

**Abstract.** *Cloud computing provides on-demand and pay-as-you-go services. Analyzing the performance of these providers is an important indicator that should be investigated before targeting the migration of a system to a cloud environment. This article aims to analyze and present the performance of IaaS, FaaS and PaaS, using the free option, in the cloud providers AWS and GCP. Based on the tested configurations, the results show that the Amazon provider performed better compared to Google.*

## 1. Introdução

O uso de plataformas de computação em nuvem é vantajoso por uma série de fatores, entre eles: escalabilidade, que é um fator primordial para aplicativos distribuídos e de alto desempenho, a facilidade de configuração das instâncias para aplicativos em execução e a eliminação do custo inicial para adquirir e operar a infraestrutura necessária [Wiechork. e Charão. 2020].

Este trabalho analisa três modelos de computação em nuvem: Infraestrutura como um serviço (IaaS), Funções como um serviço (FaaS) e Plataforma como um serviço (PaaS), onde cada modelo disponibiliza diferentes níveis de controle, gerenciamento e escalabilidade. Nos experimentos de IaaS, são utilizados *benchmarks* de CPU, memória e I/O. Em FaaS e PaaS, a análise é realizada utilizando a ferramenta de testes de desempenho JMeter. Os provedores avaliados são Amazon Web Service (AWS) e Google Cloud Platform (GCP), utilizando a opção gratuita.

A divisão desse artigo está organizado da seguinte maneira: a Seção 2 apresenta os três modelos de Computação em Nuvem utilizados nesta pesquisa. Na Seção 3, alguns trabalhos relacionados são mencionados. A Seção 4 apresenta os resultados experimentais da análise de cada serviço. Por fim, a Seção 5 conclui o trabalho e sugere direções futuras.

## 2. Computação em Nuvem

Atualmente, o termo *cloud computing* descreve a abstração de computadores, recursos e serviços baseados na web que os desenvolvedores de sistemas podem utilizar para implementar sistemas complexos. As soluções baseadas em nuvem podem ser ampliadas ou reduzidas, e as empresas, cujas soluções residem nas nuvens, pagam apenas pelos recursos que consomem [Jamsa 2013]. Com o crescimento de opções da computação em nuvem, vários modelos de implantações surgiram para atender as necessidades específicas de cada usuário. A seguir são apresentados os três modelos de computação em nuvem que foram utilizados neste trabalho, visto que cada modelo disponibiliza diferentes níveis de controle, gerenciamento e escalabilidade.

**Infraestrutura como um serviço - IaaS** Inclui os recursos de infraestrutura, desde as instalações até as plataformas de hardware que ali residem. Fornece também um conjunto de APIs, que permite a gestão e outras formas de interação com a infraestrutura por parte dos clientes. Para este artigo, os serviços utilizados são o Amazon Elastic Compute Cloud (EC2) e Google Compute Engine (GCE).

**Função como um serviço - FaaS** Também conhecido como *Serverless Computing*. A computação sem servidor permite que os desenvolvedores comprem serviços de *back-end* em uma base flexível. Na função, só é preciso configurar a quantidade de memória e o resto fica por conta do serviço alocar os recursos. A cobrança é feita pela quantidade de vezes que a função for executada. Para este artigo, as funções como serviço utilizadas são AWS Lambda e Google Cloud Functions.

**Plataforma como um Serviço - PaaS** Esse serviço permite evitar gastos, a complexidade de comprar e gerenciar licenças de software ou ferramentas de desenvolvimento, bibliotecas e outros recursos. O consumidor não gerencia a infraestrutura da nuvem subjacente. Os serviços utilizados são AWS Elastic Beanstalk e Google App Engine.

## 3. Trabalhos Relacionados

No trabalho de [Coutinho et al. 2012], é realizada uma análise de desempenho sobre um ambiente público de computação em nuvem, o HP Cloud Services, utilizando os *benchmarks* Linpack, Randmem, IOzone e Iperf, os quais utilizam métricas relativas a CPU, memória, disco e rede. Em [Farias et al. 2016], apresentam a avaliação de desempenho de um website aplicando-se o teste de carga com o auxílio da ferramenta *open source*: JMeter.

Em outro trabalho semelhante [Wiechork. e Charão. 2020], as autoras apresentam uma investigação preliminar do desempenho de consultas de banco de dados no Moodle, instalado em ambientes de nuvem da AWS e GGP. Esses experimentos foram baseados em *benchmarks* Pgbench, Sysbench e um *plugin* do próprio Moodle que analisa o desempenho do ambiente, Moodle Benchmark. No trabalho [Brebner e Liu 2010] os autores usam um conjunto de aplicativos de teste de nuvem, para analisar o desempenho e a escalabilidade de infraestruturas de nuvens reais, como Google App Engine, Amazon EC2 e Microsoft Azure. Semelhante a essa pesquisa que se insere o presente trabalho, analisando o desempenho de serviços em nuvem.

## 4. Resultados dos experimentos

A fim de mensurar o desempenho de CPU, memória e I/O, as ferramentas utilizadas na avaliação das instâncias em IaaS: Linpack, Y-cruncher e IOzone. Para os serviços de

FaaS e PaaS, foi utilizada a ferramenta JMeter. O tipo de plano de teste elaborado no JMeter para este trabalho foi configurado Grupo de Usuários, o qual é responsável por controlar o número de usuários virtuais (*threads*). Também foram adicionados elementos de configurações como: HTTP Request e Listeners. O HTTP Request define valores padrões para as requisições HTTP.

Para avaliar os resultados desses testes, os usuários foram incrementados e verificado se as respostas das requisições apresentavam algum tipo de falha. Também foi medido o tempo de resposta que o software leva para interpretar e responder ao usuário, após uma determinada requisição.

As instâncias e serviços de nuvem IaaS, FaaS e PaaS, utilizados neste trabalho, foram criados na região América do Sul para evitar o aumento da latência e possíveis resultados divergentes, se em regiões diferentes. As configurações utilizadas nas instâncias de IaaS são definidas; AWS: Intel(R) Xeon(R) CPU E5-2676 v3, 2.40 GHz, 1 GB RAM, SSD 7.7 GB. GGP: Intel(R) Xeon(R), 2.20 GHz, 1 GB RAM, HDD 9.6 GB, ambas instâncias com mesmo SO. Para os testes de FaaS e PaaS, foi criado um domínio temporário para cada serviço.

#### **4.1. Desempenho IaaS**

Nesta seção, são apresentados os resultados obtidos nos testes de CPU, memória e I/O, com base nos serviços EC2 e GCE. Nos testes, após a definição dos *scripts* para cada experimento, todos os *scripts* foram executados cinco vezes (em cada relação transação/cliente) e calculada a média dos resultados. Os resultados de cada experimento são apresentados em gráficos.

##### **4.1.1. Desempenho - CPU**

O objetivo deste experimento é avaliar se o desempenho de CPU aumenta de maneira proporcional com o aumento da capacidade. Para esse experimento foi utilizado o *benchmark* Linpack. O Linpack mede o desempenho do sistema, resolvendo equações de álgebra linear. A dimensão da matriz inicia em 100 equações, 500, e finaliza em 1000 equações. Cada dimensão, houve 5 repetições e a média de cada um é apresentada na Figura 1. O desempenho é medido em *Giga Floating Point Operations performed Per Seconds* (GFLOPS), quanto maior o resultado melhor o desempenho.

##### **4.1.2. Desempenho - Memória**

A fim de avaliar a velocidade de acesso de memória, foi utilizado o *benchmark* Y-cruncher, que é uma ferramenta que calcula o valor de  $\pi$  até determinadas casas decimais. Neste trabalho, foi iniciado com 25 milhões de algarismos, 50, e por fim, 100 milhões. O resultado é medido em segundos. Conforme observa-se na Figura 2, quanto maior os valores, menor o desempenho.

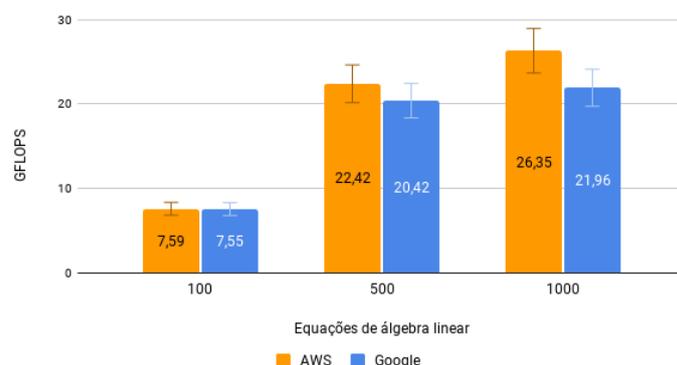


Figura 1. Resultados obtidos com o *benchmark* Linpack.

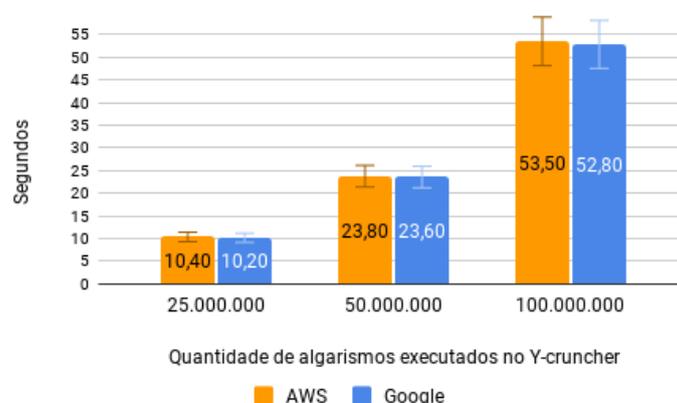


Figura 2. Resultados obtidos com o *benchmark* Y-cruncher.

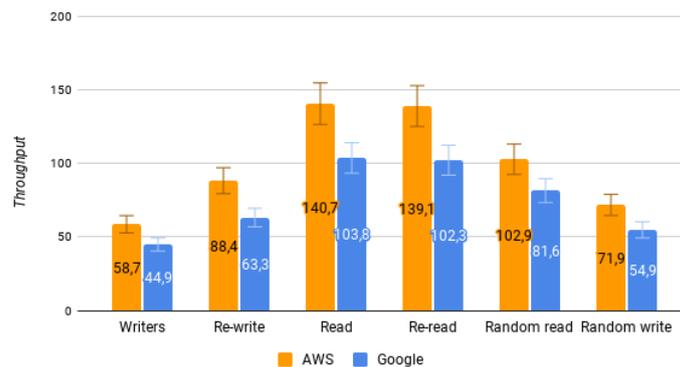
### 4.1.3. Desempenho - I/O

O objetivo deste experimento é avaliar se o *throughput* aumenta conforme a capacidade da instância é aumentada. Nesse experimento, foi utilizado o *benchmark* IOzone. O IOzone é um *benchmarking* que apresenta 13 tipos de métricas de desempenho do sistema de arquivos. Neste trabalho, utilizou-se as seguintes métricas: *write* (mede o desempenho de escrever um novo arquivo), *re-write* (mede o desempenho de gravar um arquivo que já existe), *read* (mede o desempenho da leitura de um arquivo existente), *random read* (mede o desempenho da leitura de um arquivo com acessos feitos em locais aleatórios no arquivo), *random write* (mede o desempenho da escrita de um arquivo com acessos feitos em locais aleatórios no arquivo).

Na Figura 3 os testes do IOzone são apresentados, quanto maior os valores, melhor o desempenho. É importante ressaltar que a instância da AWS possui SSD, enquanto que no Google, HDD. Se tratando de desempenho, o SSD garante mais velocidade, seja para iniciar um sistema operacional, abrir programas ou arquivos, em comparação ao HD convencional.

## 4.2. Desempenho FaaS

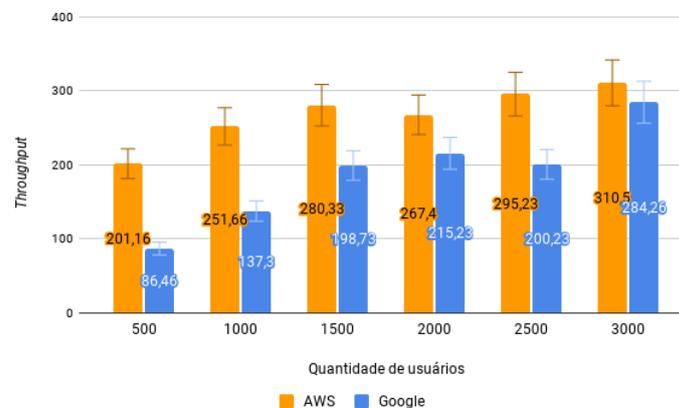
Nos testes em FaaS, foram utilizados inicialmente 500 usuários virtuais simultâneos. Após isso, o número de usuários foi dobrado, finalizando em 3000 usuários ativos. Cada usuário



**Figura 3. Resultados obtidos com o IOzone.**

executa uma requisição e aguarda a resposta. O tempo de inicialização foi de 0,1 (segundos) e as interações (quantidade de vezes que foi executado no teste) definido em 1.

Para o teste realizado, como mostra a Figura 4, foi utilizado o componente Gráfico Agregado que calcula a média do tempo de resposta para cada requisição. Para avaliar os resultados destes testes, os usuários foram incrementados e verificado se as respostas das requisições apresentam algum tipo de falha na ferramenta. Em cada intervalo de usuários, as requisições foram repetidas cinco vezes e calculado a média dos resultados. A métrica levada em consideração foi o *throughput*, quanto maior o valor melhor é o desempenho.



**Figura 4. AWS Lambda X Google Cloud Functions.**

### 4.3. Desempenho PaaS

Nos testes em PaaS foram utilizados inicialmente 500 usuários virtuais simultâneos, fazendo requisições HTTP nos endereços das aplicações. Após isso, o número de usuários foi dobrado até chegar em 3000 usuários ativos. Cada usuário executou uma requisição e aguardou a resposta. O tempo de inicialização foi de 0,1 (tempo em segundos) e as interações (quantidade de vezes que foi executado o teste) definido em 1. As configurações para todos os tipos de testes são iguais, o que difere é a quantidade de usuários simulados, que variam de 500 a 3000, e o tipo de teste usado.

A Figura 5 apresenta a comparação dos provedores em cada intervalo, onde a execução foi realizada três vezes e calculada a média aritmética. A métrica calculada foi o

*throughput*. Neste experimento, com exceção do segundo, os resultados do Google App Engine foi um pouco melhor que os da AWS Elastic Beanstalk.

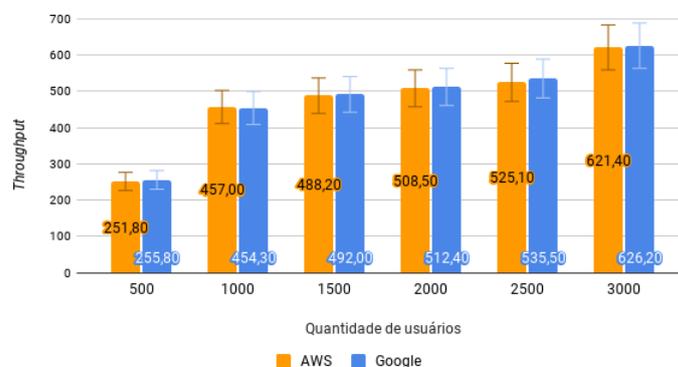


Figura 5. AWS Elastic Beanstalk X Google App Engine.

## 5. Conclusão e Trabalhos Futuros

Com as ferramentas de testes de desempenho e os *benchmarks* utilizados neste trabalho, foi possível avaliar o desempenho de IaaS, FaaS e PaaS, utilizando o plano gratuito. De acordo com os dados coletados, é notável que o provedor Amazon apresentou melhor desempenho na grande maioria dos resultados, em comparação com o provedor Google. No entanto, esses resultados podem sofrer mudanças dentro de certos limites, optando por planos de serviços pago. Baseando-se no conhecimento adquirido através da pesquisa desse trabalho, sugerem-se como recomendações para trabalhos futuros: utilizar outros *benchmarks* e ferramentas de testes de desempenho, realizar a avaliação em outros provedores de nuvens e se houver recursos financeiros, avaliar esses provedores com outros planos.

## Referências

- Brebner, P. e Liu, A. (2010). Performance and cost assessment of cloud services. volume 6568, pages 39–50.
- Coutinho, E., Rego, P., Gomes, D., e Souza, J. (2012). Análise de desempenho com benchmarks em um ambiente público de computação em nuvem.
- Farias, E. M. B., Sousa, E. S., e Nascimento, R. P. (2016). Avaliação de desempenho de um website utilizando apache jmeter: Um estudo de caso do website institucional da universidade federal do oeste do pará. SBRT.
- Jamsa, K. (2013). *Cloud Computing: SaaS, PaaS, IaaS, Virtualization, Business Models, Mobile, Security and More*. Burlington, MA : Jones & Bartlett Learning.
- Wiechork., K. e Charão., A. (2020). Investigating the performance of moodle database queries in cloud environments. In *Proceedings of the 22nd International Conference on Enterprise Information Systems - Volume 1: ICEIS*, pages 269–275. INSTICC, SciTePress.