

# Análise do Custo-benefício de Funções como Serviço e Infraestrutura como Serviço

Waliff Cordeiro Bandeira<sup>1</sup>, Leonardo Rebouças de Carvalho<sup>2</sup>,  
Aleteia Patricia Favacho de Araujo<sup>3</sup>

<sup>1</sup>Departamento de Ciência da Computação - Universidade de Brasília (UnB) - Brasília - DF - Brazil

**Abstract.** *Function as a Service (FaaS) offers high performance in its applications to be used with high availability and payment based on demand. These features, however, come with a high added monetary value, which makes it not feasible to use this service full-time for various types of applications and contexts. To better understand this inflection point, it is necessary to test and compare, in a practical way, what is the acceptable threshold to choose a service among other options, before starting the application development. For this, Infrastructure as a Service (IaaS) will be used, which is a service that provides computing resources, IaaS was selected in order to compare and analyze the cost-benefit at the expense of FaaS.*

**Resumo.** *O Function as a Service (FaaS) oferece alto desempenho em suas aplicações para ser utilizado com alta disponibilidade e pagamento baseado na demanda. Esses recursos, entretanto, vêm com um alto valor monetário agregado, o que faz com que não seja viável a utilização desse serviço em tempo integral para vários tipos de aplicações e contextos. Para melhor entender esse ponto de inflexão, faz-se necessário testar e comparar, de forma prática, qual o limiar aceitável para escolher um serviço dentre outras opções antes de iniciar o desenvolvimento da aplicação. Para isso será utilizada a Infraestrutura como Serviço (IaaS), que é um serviço de disponibilização de recursos computacionais, o IaaS foi escolhido a fim de comparar e analisar o custo-benefício em detrimento do FaaS.*

## 1. Introdução

Atualmente, o *Function as a Service* (FaaS) - em português, Função como Serviço - está presente na realidade de diversos desenvolvedores. Essa tecnologia permite a criação de funções que retornam o resultado esperado em um curto espaço de tempo, em uma interface simples e de fácil acesso, disponível na maior parte dos grandes provedores de serviço em nuvem como AWS, Azure e GCP. Logo, o FaaS tem sido visto como um serviço de extremo valor para diversas aplicações que possuem o tempo como um fator decisivo em seu desempenho.

Segundo o artigo [Sewak and Singh 2018], a utilização de FaaS cresce em um ritmo de 32,7% ao ano, o que evidencia a necessidade de um melhor entendimento sobre o custo da utilização desse serviço, levando-se em consideração seu alto desempenho e disponibilidade mas também ao considerar seu alto custo agregado. Para essa análise comparativa, é importante considerar outros serviços que oferecem a capacidade de processamento em nuvem para que se possa compreender o custo-benefício desses serviços oferecidos pelos

provedores de nuvem.

Assim, optou-se por comparar o FaaS com o IaaS, que é o serviço que possibilita pessoas e empresas a utilizarem recursos de hardware de forma remota, proporcionando uma infraestrutura computacional sem a necessidade de aquisição de máquinas físicas. Nesse contexto o EC2 foi escolhido para representar a IaaS na comparação com outros três serviços de FaaS que são o Lambda da AWS, Cloud Functions da GCP e o Function App da Azure. Diante do exposto, este trabalho propõe uma análise entre as duas diferentes interfaces de disponibilização de serviços, a IaaS e a FaaS, de forma a estudar o melhor custo-benefício entre eles nos três principais provedores de nuvem pública AWS, Google Cloud e Azure.

## 2. FaaS e IaaS

O *Function as a Service* é a categoria de serviço de computação em nuvem orientada a eventos, ou seja, fornece uma interface simples para implementação e execução de funções. Como citado, as funções são executadas por eventos (gatilhos), os *triggers*, que pode ser, por exemplo, métodos HTTP. O FaaS está ganhando espaço no mercado por se mostrar escalável, com alta disponibilidade e desempenho. Exemplos de serviços FaaS são o Lambda da AWS, Cloud Function da GCP e Azure Functions da Azure. A *Infrastructure as a Service*, por sua vez, fornece estrutura computacional de forma facilitada, ou seja, através dos provedores de nuvem é possível contratar máquinas em variadas regiões te dando acesso de forma remota ao poder computacional que elas têm a oferecer. A IaaS apresenta flexibilidade de configuração, capacidade para escalabilidade e previsibilidade no valor a ser pago com a contratação do serviço. Exemplos de serviços IaaS são o EC2 da AWS, Azure Virtual Machines da Azure e o Google Compute Engine da GCP.

## 3. Experimento

A parte experimental consiste em proporcionar condições semelhantes de teste com o objetivo de garantir máxima precisão nas análises e conclusões obtidas sobre a relação custo-benefício do uso do FaaS versus IaaS. É importante ressaltar que a complexidade de implementação e uso está sendo desconsiderada, visto que foca-se no custo monetário em comparação ao poder computacional disponibilizado.

Para estressar as funções e comparar como ambos os serviços performam em determinadas condições de uso, foram escolhidas funções utilizadas pelo artigo [Maissen et al. 2020]. O artigo citado utiliza um *benchmark*, testa as funções e disponibiliza os resultados em relação ao FaaS. Dessa forma, os testes podem ser realizados em ambiente de nuvem através dos serviços de FaaS e IaaS nos três provedores.

### 3.1. Ambientes similares para os testes

A partir de interfaces diferentes para utilização do FaaS e IaaS, o *benchmark* utilizado [Maissen et al. 2020] não pode, de forma simples, ser reaproveitado no ambiente IaaS. Dessa forma, foi desenvolvida uma *Application Programming Interface* (API) para cada linguagem que será utilizada nos testes. A API foi desenvolvida neste trabalho com o objetivo de agregar o menor *overhead* possível, de modo a tornar o teste mais preciso. Assim sendo, foram desenvolvidas APIs nas linguagens Golang, Python e Nodejs.

Dessa forma, com as APIs criadas, os testes de FaaS e IaaS serão realizados de maneira semelhante quanto a interface, pois ambos poderão ser estressados pela utilização de métodos HTTP.

### 3.1.1. Benchmark

O Apache JMeter [Erinle 2017] é uma ferramenta consolidada para testes de carga e performance, útil para estressar aplicações. Portanto, pretende-se avaliar como a aplicação se comporta com um aumento considerável nos números de requisições, que visa validar a qualidade e a disponibilidade da aplicação em ambiente de estresse.

Para avaliação do custo-benefício, será necessário também triangular essas informações com o valor monetário gerado pelo teste.

- IaaS: o valor do serviço é baseado nas configurações das máquinas contratadas e o valor final é baseado no tempo em que o serviço se manteve contratado, de forma que seja possível contratar máquinas mais ou menos robustas e por maior ou menor quantidade de tempo. Porém, mesmo que a máquina não esteja sendo utilizada, se estiver contratada, o valor será cobrado.
- FaaS: o valor do serviço é baseado na demanda, ou seja, mesmo que a função esteja criada, o valor gerado será baseado apenas no tempo de execução da função e no número de requisições.

Ao ter diferentes formas de precificação, torna-se ainda mais difícil para o usuário decidir qual é o melhor serviço para o seu contexto. Assim sendo, se o usuário estiver buscando um alto desempenho em um curto espaço de tempo, o FaaS pode ser a opção a se contratar. Todavia, para um caso mais geral, faz-se a análise dos resultados obtidos através dos testes para possibilitar a simplificação da escolha do usuário, com o objetivo de tornar perceptível quando é mais vantajoso contratar um serviço em detrimento ao outro.

## 4. Conclusão

FaaS oferece diversas vantagens ao utilizá-lo, tais como um alto desempenho, alta disponibilidade e fácil interface para implementação. O IaaS, de forma comparativa, tende a oferecer um menor desempenho, uma maior carga de trabalho para configuração do ambiente e um custo mais previsível. Para tanto, torna-se necessário uma forma de melhor avaliar o custo-benefício da utilização desse serviço. Os testes propostos neste trabalho tendem a facilitar essa escolha para o usuário, proporcionando uma melhor compreensão da sua real necessidade e demanda em comparação ao custo aceitável para o desenvolvimento.

## Referências

- Erinle, B. (2017). *Performance Testing with JMeter 3*. Packt Publishing.
- Maissen, P., Felber, P., Kropf, P. G., and Schiavoni, V. (2020). Faasdom: A benchmark suite for serverless computing. *CoRR*, abs/2006.03271.
- Sewak, M. and Singh, S. (2018). Winning in the era of serverless computing and function as a service. In *2018 3rd International Conference for Convergence in Technology (I2CT)*, pages 1–5.