

# Utilizando o OpenFaaS em plataformas embarcadas e server

Igor C. D. de Almeida, Guilherme Galante, Marcio S. Oyamada

Centro de Ciências Exatas e Tecnológicas - Colegiado de Ciência da Computação –  
Universidade Estadual do Oeste do Paraná (UNIOESTE)

{igor.almeida6, guilherme.galante, marcio.oyamada}@unioeste.br

**Resumo.** *Este trabalho compara a computação em nuvem e na borda usando OpenFaaS. Uma função de verificação de números primos foi executada em um servidor e na plataforma embarcada S905x (ARM A53). A borda é viável para cargas leves, mas torna-se um gargalo para cálculos complexos. O estudo destaca a importância do balanceamento dinâmico entre nuvem e borda visando otimizar o desempenho e escalabilidade.*

## 1. Introdução

A computação em nuvem (*cloud computing*) é um paradigma que permite acesso a recursos computacionais de forma remota, por custos muito baixos, geralmente em um modelo *pay-per-use* [Rajaraman 2014].

O paradigma *Serverless*, também conhecido como *Function as a Service* (FaaS) [Shafiei *et al.*, 2022], é um modelo de desenvolvimento em nuvem que se refere ao conceito de construir e executar aplicações que não necessitam de gerenciamento de servidor [Cloud Native Computing Foundation 2018]. Nesse modelo, os desenvolvedores implementam funções orientadas a eventos, enquanto a plataforma FaaS gerencia automaticamente a alocação de recursos e escalabilidade [Agudelo-Sanabria e Jindal 2021].

A computação na borda (*edge computing*) busca aproximar a fonte dos dados do seu processamento, possibilitando análises e respostas em tempo real e mitigando desafios da computação em nuvem, como os problemas de segurança, latência elevada e alto consumo de energia [Hong e Wang 2018]. Porém, os desenvolvedores precisam gerenciar os recursos, escalabilidade e balanceamento de carga [Xie *et al.* 2021]. Enquanto a implementação da FaaS na nuvem é escalável e pode se beneficiar da infraestrutura robusta, o uso de dispositivos na borda oferece maior privacidade e redução na latência e consumo de energia.

A integração entre computação em borda e *Serverless* apresenta potencial, mas ainda enfrenta desafios. Diante disso, este trabalho tem como objetivo avaliar a integração e comparar o FaaS executando na nuvem e em dispositivos na borda. Para isso, foram analisadas métricas de desempenho de uma mesma função executada tanto em uma plataforma embarcada, representando a borda, quanto em um servidor, representando a nuvem, utilizando o OpenFaaS (2025), especificamente o modelo faasd, um contêiner que encapsula todos os recursos necessários para fornecer o serviço FaaS sem a necessidade de um gerenciador de contêineres.

## 2. Metodologia

Para a realização dos experimentos, foram utilizados dois dispositivos: um para executar a função de servidor e uma plataforma embarcada. O servidor utilizado possui um processador AMD Ryzen 7 5700G e 64GB de memória RAM, utilizando o Linux distribuição Ubuntu. A plataforma embarcada consiste em uma TVBox Tigre TG3, equipada com um SoC S905x composto por um QuadCore ARM Cortex A53 1.5GHz com 2GB de memória RAM. O processador QuadCore é equivalente ao Raspberry PI 3B. A TVBox foi doada pela Receita Federal, sendo fruto de apreensões em fiscalização, e faz parte de um projeto de reutilização desses equipamentos, sendo descaracterizadas para executar o Linux distribuição Armbian. Para os testes, as requisições são realizadas a partir de um notebook, com um processador Intel Core i5-13240H 2,10 GHz, com 16 GB de memória RAM utilizando Linux Ubuntu, e todos os dispositivos estão na mesma rede cabeada de 10 Mbps, com os pontos conectados via *switch*. Os testes foram realizados com a rede local ativa, com tráfego de outros dispositivos presentes, sem a implementação de técnicas de isolamento de tráfego.

Para os testes, foi implementada uma função no OpenFaaS em Python para verificação de números primos, juntamente com um *script* para execução no cliente responsável pelo envio de requisições HTTP e medição do tempo de resposta, utilizando a biblioteca *requests*. No entanto, é importante destacar que os testes foram conduzidos em um ambiente específico, podendo haver variações nos resultados em diferentes cenários devido a fatores como rede, carga do sistema e limitações do hardware utilizado. O monitoramento do consumo de memória, que é exibida de forma cumulativa, foi realizado por meio do containerd, plataforma utilizada pelo faasd para a execução de contêiners. Em ambos os dispositivos, foi empregada a mesma imagem Docker, ajustando o endereço de destino das requisições.

Para os testes, também foi implementado um *script*, responsável por todo o controle, para invocar a função na plataforma embarcada (TG3) e, caso a resposta demore mais de 30 segundos, tempo limite definido para este experimento, a requisição de invocação é redirecionada ao servidor.

## 3. Resultados

Todos os experimentos foram realizados oito vezes, e os valores apresentados correspondem às médias obtidas. O primeiro experimento realizado avaliou o consumo de memória dos dispositivos durante a disponibilização da função para uso. O servidor utilizou 2,237 MB de memória, enquanto a TVBox TG3 consumiu 1,573 MB, usando uma menor quantidade de memória por ser uma imagem específica para ARM, adaptado pelo próprio OpenFaas.

Em seguida, foram realizadas chamadas de teste com um número pequeno (valor 7). O servidor processou a requisição em 0,0773 segundos, consumindo 0,0611 MB adicionais. Em comparação, a TG3 levou 0,6756 segundos e utilizou 0,0672 MB demonstrando um desempenho inferior tanto em tempo de resposta quanto em consumo de recursos, conforme apresentado nas Figuras 1 e 2.

Posteriormente, foi testado um número primo maior, o 281474976710597. O servidor respondeu em 0,8362 segundos, consumindo 0,046 MB enquanto a TG3 demorou 18,4534 segundos e utilizou 0,0365 MB.

Para avaliar o redirecionamento de requisições, foi utilizado o número 1125899906842597. A TG3 não conseguiu processar a requisição dentro do limite de

tempo 30 segundos definido no experimento, enquanto o servidor concluiu a requisição em apenas 1,6067 segundos.

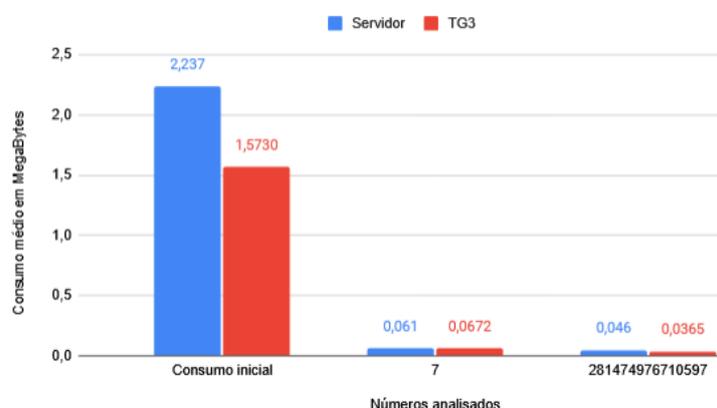


Figura 1. Comparação de consumo médio de memória em MB

Os testes demonstram que, para números menores, a diferença de desempenho entre o servidor e a TVBox é menos significativa. No caso do número 7, a TG3 consumiu menos memória para execução, com uma diferença de tempo de resposta médio de 0,5983 segundos (8,74 vezes mais lenta que o servidor). Entretanto, para o número 281474976710597, a diferença foi de 22,07 vezes. No teste de redirecionamento, a TG3 não conseguiu fornecer uma resposta dentro do tempo limite, enquanto o servidor processou a requisição rapidamente, demonstrando a diferença na capacidade de processamento dos dois dispositivos.

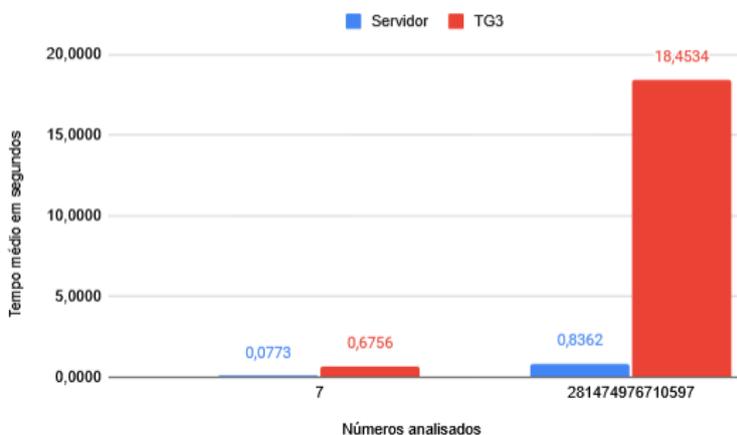


Figura 2. Comparação do tempo médio de resposta (em segundos)

#### 4. Conclusão e Trabalhos Futuros

O paradigma *serverless* possibilita a execução de funções sob demanda, sem a necessidade de gerenciar infraestrutura, otimizando alocação de recursos e escalabilidade e com custos reduzidos ao usuário.

Nesse contexto, este trabalho teve como objetivo comparar o uso de FaaS na computação em nuvem e na borda, estudando um caso de uso, permitindo que métricas fossem medidas e possibilitando a análise das vantagens do uso de cada paradigma.

Os experimentos realizados demonstraram que a execução de funções na TVBox e no servidor apresentaram diferenças significativas no tempo de resposta e consumo de

recursos. Enquanto a TG3 mostrou-se capaz de lidar com cargas menores, seu desempenho foi consideravelmente inferior ao servidor conforme a necessidade de processamento aumentou.

Os resultados mostraram que, para cargas leves, a execução na borda é viável, pois o aumento no tempo de resposta é pequeno e o consumo de recursos é baixo. No entanto, com cargas maiores, a borda se torna um gargalo, tornando o redirecionamento para o servidor mais eficiente. A implementação desse mecanismo validou a abordagem híbrida, evidenciando que a combinação entre borda e processamento centralizado pode otimizar aplicações *Serverless*. Além disso, é importante considerar a influência da latência de rede em cenários de produção, pois isso pode aumentar o tempo de resposta, principalmente no contexto de nuvem.

Para trabalhos futuros, planeja-se a análise de diferentes tipos de aplicações, como processamento de imagens e outros, com variados tempos limites de execução, a fim de ampliar o entendimento sobre o paradigma *serverless* borda-nuvem. Além disso, serão avaliadas outras métricas como latência e consumo de energia por requisição. Também serão utilizados para teste serviços como o Google Cloud e Amazon Web Services e será analisado o desempenho do FaaS.

Assim, este estudo contribui para a compreensão das limitações e vantagens da computação *serverless* na borda e na nuvem e reforça a necessidade de um balanceamento dinâmico entre os diferentes ambientes de execução.

## Referências

- Hong, X. e Wang, Y. (2018) "Edge Computing Technology: Development and Countermeasures", *Strategic Study of CAE*, 20(2): 20–26. Disponível em: <https://www.engineering.org.cn/sscae/EN/10.15302/J-SSCAE-2018.02.004>
- Documentação OpenFaaS (2025). Disponível em: <https://docs.openfaas.com/>
- Shafiei, H; Khonsari, A. e Mousavi, P. (2022) Serverless Computing: A Survey of Opportunities, Challenges, and Applications. *ACM Comput.* Disponível em: <https://doi.org/10.1145/3510611>
- Agudelo-Sanabria, S. D.; Jindal, A. (2021) The ifs and buts of the development approaches for iot applications. Disponível em: <http://arxiv.org/abs/2101.09796>.
- Cloud Native Computing Foundation. (2017) CNCF WG-Serverless Whitepaper v1.0. Disponível em: [https://gw.alipayobjects.com/os/basement\\_prod/24ec4498-71d4-4a60-b785-fa530456c65b.pdf](https://gw.alipayobjects.com/os/basement_prod/24ec4498-71d4-4a60-b785-fa530456c65b.pdf)
- Xie, R., Tang, Q., Qiao, S., Zhu, H., Yu, F. R. e Huang, T. (2021) "When Serverless Computing Meets Edge Computing: Architecture, Challenges, and Open Issues", *IEEE Wireless Communications*, 28(5): 40-47. Disponível em: <https://ieeexplore.ieee.org/abstract/document/9756233>.