# Avaliação da computação na borda/névoa utilizando *serverless* em sistemas IoT

**Rafael Pascoali Czerniej, Marcio Seiji Oyamada**

Centro de Ciências Exatas e Tecnológicas - UNIOESTE - Campus Cascavel

{rafael.czerniej, marcio.oyamada}@unioeste.br

***Resumo.*** *Este artigo tem como objetivo avaliar o modelo de computação serverless utilizando dispositivos na borda/névoa. O modelo serverless pode facilitar a disponibilização de serviços IoT pois permite executar código sem a necessidade de gerenciamento da infraestrutura. Um estudo de caso de processamento de imagens está sendo implementado, com a utilização de uma Raspberry PI 3 para execução do ambiente serverless OpenFaaS.*

## 1. Introdução

Aplicações *Internet of things* (IoT) têm se popularizado recentemente em diversos domínios, porém enfrenta desafios na integração de uma variedade de componentes de hardware e software, que são inerentemente distribuídos e que podem também ser interdependentes, o que leva a alta complexidade, um longo tempo de desenvolvimento e dificuldades de manutenção [Udoh e Kotonya 2017].

Uma solução para o desenvolvimento de aplicações mais eficientes e com menor tempo de desenvolvimento é o paradigma de *Function as a service* (FaaS) ou *serverless*, no qual problemas como o funcionamento e configuração de servidores são abstraídos pela plataforma, permitindo um maior foco na lógica das funções criadas pelo desenvolvedor [Hassan *et al* 2021]. A ativação das funções é realizada sob demanda, conforme eventos detectados pelos dispositivos, com escalabilidade dinâmica, fornecendo desempenho e maior tolerância a falhas [Agudelo-Sanabria e Jinda 2021].

Dispositivos IoT normalmente geram uma grande quantidade de dados que devem ser processados rapidamente, porém processar tais dados na nuvem pode levar a problemas de latência nas respostas. Para resolvê-los, uma possibilidade é utilizar dispositivos na névoa para processar os dados em um servidor local ou mais próximo dos dispositivos IoT [Abhishek Hazra *et al* 2023].

Neste estudo é avaliado a utilização do FaaS, mais especificamente a solução OpenFaaS executada na plataforma embarcada Raspberry Pi 3. Um estudo de caso de um sistema de contagem de faces acionado por detecção de movimento no cliente foi utilizado para avaliar a execução do OpenFaaS em funções complexas, com dependências de bibliotecas externas como o OpenCV. Este estudo servirá como base para avaliar o compromisso entre requisitos como desempenho, consumo de energia, latência e custos, comparando a execução na borda em relação à dispositivos na névoa e na nuvem.

## 2. Metodologia

A plataforma OpenFaas [Alex Ellis 2021], é um provedor de *function as a service* de código aberto usando containers Docker, disponibilizando a execução de tais funções para dispositivos na rede, podendo retornar resultados ao fim do processamento.

Neste estudo foi utilizada a versão Faasd do OpenFaas, uma reimplementação mais leve desenvolvida com casos de uso IoT em mente, utilizando *Containerd* e *Container Network Interface* para executar funções localmente dentro de containers Docker. Neste trabalho o Faasd foi instalado em uma placa Raspberry pi 3 B conectada à uma rede local via conexão cabeada 10Mbps. Os experimentos foram realizados com a rede ativa, com tráfego concorrendo com outros dispositivos na rede.

A aplicação cliente é executada em uma TvBox descaracterizada. Esse equipamento faz parte de uma ação da Receita Federal para reutilização de materiais apreendidos. A TvBox possui uma plataforma S905X, contando com 2Gb de memória RAM e o processador QuadCore ARM Cortex A53, formatada com um sistema operacional Armbian OS 24.11.0 Noble, em conjunto com a câmera Microsoft Lifecam HD-5000.

O estudo de caso realizado neste artigo é uma implementação de um sistema de detecção e contagem de faces, instalado nos corredores do bloco de laboratórios pertencente ao curso de Ciência da Computação - Unioeste. A detecção das faces é realizada utilizando um Código em Python do algoritmo Haar Cascade Classifier [OpenCv 2001], disponível na biblioteca OpenCV.

Na TvBox, o *software* Motion foi utilizado para coleta de imagens do corredor, salvando a captura da imagem caso haja alguma detecção de movimento. A aplicação responsável por processar as imagens foi escrita em Python. O fluxo de execução é apresentado na Figura 1. A função de detecção de faces é implementada tanto localmente (TvBox), quanto na Raspberry Pi 3 utilizando o OpenFaaS. Caso o uso médio de CPUs na TvBox seja menor que 70%, a imagem é processada localmente, caso contrário o Faasd é requisitado para a imagem ser processada na Raspberry Pi. Essa abordagem permite que a carga de trabalho seja distribuída entre os dispositivos na borda, podendo ser estendida para execução em recursos alocados na nuvem futuramente, Os códigos desenvolvidos neste projeto estão disponíveis no GitHub[1].

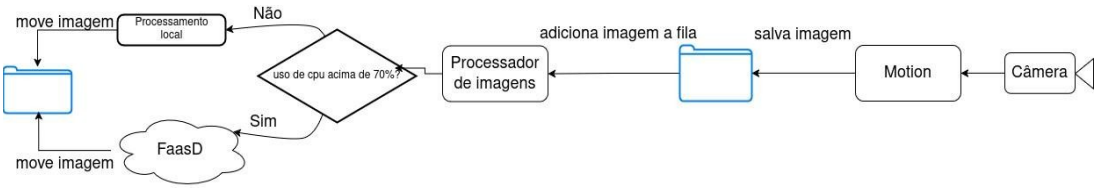


**Figura 1. diagrama demonstrando o estudo de caso**

Os dados de contagem de faces são enviados para a nuvem, no serviço Thingspeak [Lee Lawlor et al 2011], para visualização dos dados em tempo real (Figura 2). O *link* do canal utilizado está disponível no endereço da ferramenta[2]. Para

1 https://github.com/RafaelPasCz/erad_openfaas

2 https://thingspeak.mathworks.com/channels/2790027

coletar métricas de desempenho e uso de dados, o *software* Prometheus foi utilizado, coletando o uso médio de CPU, *Download* e *Upload* de dados da TvBox e do Raspberry Pi.

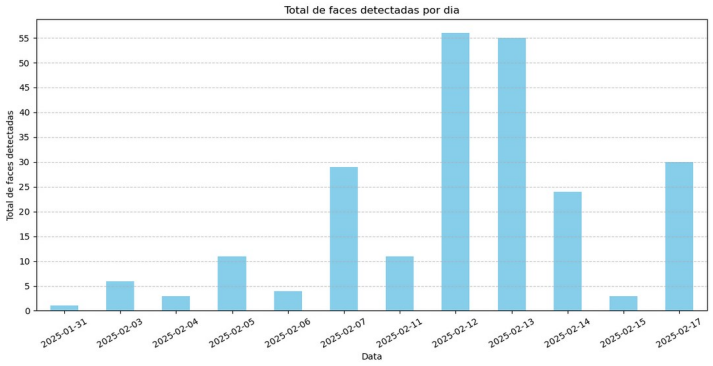


**Figura 2 - Dados apresentados no Thingspeak**

## 3 - Resultados preliminares

Os testes estão atualmente em andamento, as Figuras 3 e 4 representam estatísticas de um caso em que houve uma simulação de um pico de quatro requisições em rápida sucessão para a função no OpenFaaS executando na Raspberry Pi, resultando em um rápido crescimento de recepção de dados seguido de um alto uso da CPU.

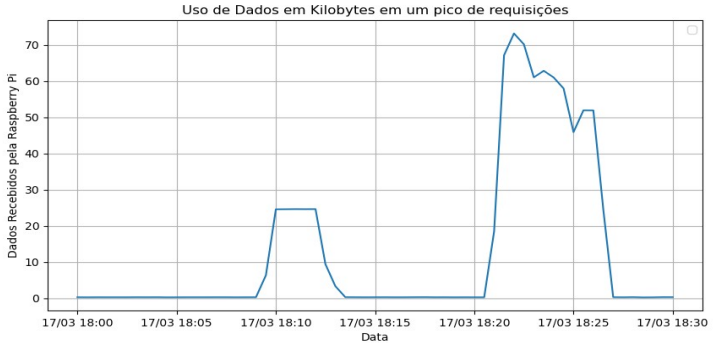


**Figura 3. Recepção de dados na Raspberry PI (Kb)**

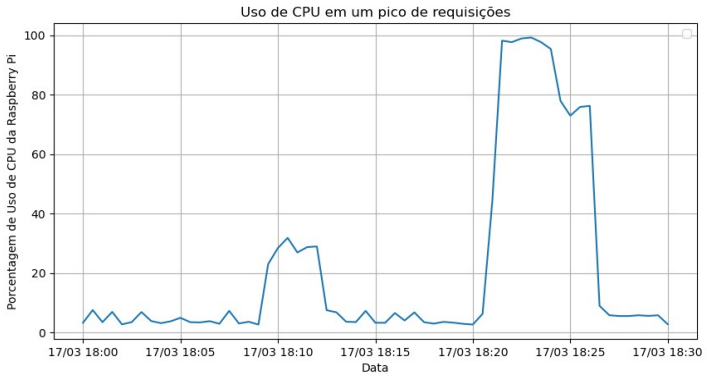


**Figura 4. Uso de CPU na Raspberry pi**

## 4 - Conclusão e trabalhos futuros

Dos testes iniciais, observamos que a utilização de *Function as a Service* em aplicações IoT é uma opção viável para a implementação de processamento na borda. No estudo de caso, o encapsulamento de uma função com dependências de bibliotecas externas (OpenCV) foi realizada, sendo possível sua execução. Como não existe a necessidade de gerenciamento da infraestrutura, o cliente deve se preocupar apenas em chamar a função e obter o resultado.

Trabalhos futuros devem explorar análises detalhadas de desempenho e avaliar a eficácia da execução adaptativa utilizando processamento local, na névoa e na nuvem. Adicionalmente, será realizado testes envolvendo a implementação do FaaS em dispositivos mais potentes que a Raspberry para a execução das funções, mas na mesma rede, simulando o processamento na névoa. Além do desempenho serão avaliados critérios como latência e consumo de energia por requisição em dispositivos na borda e na névoa.

## Referências


Udoh, I. S e Kotonya, G. (2017) “Developing IoT applications: challenges and frameworks.” Iet Cyber-Phys. Syst., Theory Appl. v. 3, n. 2, p. 65-72.

Hassan, B. H et al. (2021) "Survey on serverless computing." Journal of Cloud Computing 10 : p. 1-29.

Agudelo-Sanabria, S. D e Jindhal, A. (2021) “The ifs and Buts of the Approaches for IoT Applications” arXiv preprint arXiv:2101.09796.

Hazra, A et al (2023) “Fog computing for next-generation Internet of Things: Fundamental state-of-the-art and research challenges”. Computer Science Review, 48, 100549.

Ellis, A. (2021) “serverless for everyone else”. Disponivel em https://store.openfaas.com/l/serverless-for-everyone-else

Lee Lawlor et al (2011) “Thingspeak” disponível em https://github.com/iobridge/thingspeak/blob/master/README.textile

Huamán, A (2023) “Opencv:Cascade Classifier”. Disponível em https://docs.opencv.org/4.9.0/db/d28/tutorial_cascade_classifier.html