MicroFaaS on OpenFaaS: An Embedded Platform for Running Cloud Functions

Abin B. George†, Anthony Byrne† and Ayse K. Coskun†
†Boston University, Boston, MA 02215; Emails: {abg309, abyrne19, acoskun}@bu.edu

***Abstract*—Function-as-a-Service (FaaS) platforms present a new cloud computing paradigm by enabling serverless deployment and execution of functions. MicroFaaS, a cost-effective and energy-efficient datacenter architecture, replaces x86-based rack servers with ARM-based single-board computers (SBCs). This paper focuses on enhancing MicroFaaS by incorporating OpenFaaS, a popular secure function building and deployment framework. Through extensive experimentation, *MicroFaaS on OpenFaaS* showcases improved energy efficiency, ease-of-use, and scalability over traditional cloud systems.**

***Index Terms*—serverless, function-as-a-service, single-board computers, OpenFaaS**

I. Introduction

The success of cloud computing has been primarily driven by the desire to increase efficiency. This has led to the development of function-as-a-service (FaaS), a cloud computing paradigm in which developers can deploy stateless functions to be hosted and managed by a cloud platform.

Current cloud computing hardware implementations rely on traditional x86 CPUs, despite their poor energy efficiency compared to modern alternatives. The result is that cloud computing datacenters consumed over 1% of all electricity used globally in 2020 [1], [2]. Previously, we introduced *MicroFaaS*, an alternative hardware model that breaks datacenters free from their traditional reliance on x86-based rack servers [3]. In this new iteration, we propose *MicroFaaS on OpenFaaS*, which uses the same hardware model of *MicroFaaS* but adds an OpenFaaS plug-in that allows users to more easily access and run FaaS workloads on *MicroFaaS*.

MicroFaaS on OpenFaaS[1] is a FaaS-focused datacenter hardware model that executes FaaS functions on ARM-based single-board computers (SBCs) via OpenFaaS. The use of ARM-based SBCs provide a cheaper, more secure, and more energy-efficient alternative than x86 rack server models by taking advantage of the stateless and granular nature of serverless functions.

The key difference between the two iterations of *MicroFaaS* is the orchestration platform. The previous version used proof-of-concept cluster orchestration software to demonstrate feasibility. However, in this paper, OpenFaaS, a popular open-source FaaS framework, is incorporated as the orchestration platform for *MicroFaaS*. This integration enhances the deployment and execution of functions on *MicroFaaS*, making it easier for cloud platform users. The purpose of incorporating OpenFaaS is to provide an updated model of MicroFaaS that aligns with the current framework for FaaS workloads.

[1]*MicroFaaS on OpenFaaS* is open-source and available to the community at github.com/peaclab/MicroFaaS.

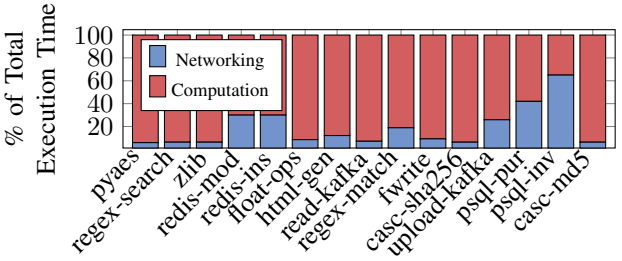

Figure 1: Function Profiling

II. Methods

Our *MicroFaaS* prototype utilizes multiple BeagleBone Black SBCs, each containing a single-core 1GHz ARM Cortex-A8 microprocessor. The *MicroFaaS* prototype includes an orchestrator SBC, which is responsible for turning on the available worker node, and 10 worker SBCs. Each SBC worker node runs serverless functions using a MicroPython interpreter, following a single-tenant, run-to-completion model. Details on optimizing the minimal OS for worker nodes and execution model of the worker nodes can be found in our previous paper [3].

The main difference in our new *MicroFaaS* version is the adoption of OpenFaaS as the orchestrator. OpenFaaS simplifies the deployment and invocation of serverless functions for cloud computing users. A user interacts with an OpenFaaS gateway through its CLI or web interface, which directs requests to the *MicroFaaS* cluster. Using the OpenFaaS provider API, *MicroFaaS* sends a JSON object containing the function name along with the function parameters to invoke functions on worker nodes. Workers execute functions with provided inputs and return outputs via the OpenFaaS gateway. After 15 seconds of inactivity, workers power off to conserve energy.

In order to verify the feasibility of *MicroFaaS* as an energy-efficient alternative to conventional serverless platforms, we evaluate the *MicroFaaS* system against an x86-based rack server (AMD 12-core processor) by running 1,000 invocations of a series of workload functions (Table I). To compare the two cloud computing platforms fairly, we ran the workload functions via OpenFaaS on the x86 server with 1, 6, and 12 threads. We also conducted function profiling using utilities **nload** and **operf** to characterize the functions as either "CPU Intensive" or "Network Intensive" (Figure 1).

III. Evaluation Results

Experimental results show that *MicroFaaS on OpenFaaS* is more energy-efficient than conventional rack servers, with an average 50% reduction in energy consumption per function.

TABLE I: Workload Functions

Name	Description	Name	Description
FloatOps*	floating-point trigonometric operations	RedisInsert	insert Redis key-value record
CascSHA	cascading SHA256 hash calculations	RedisUpdate	update Redis key-value record
CascMD5	cascading MD5 hash calculations	SQLSelect	query our PostgreSQL server using SELECT
HTMLGen	dynamically generate and serve HTML	SQLUpdate	query our PostgreSQL server using UPDATE
Decompress*	extract a DEFLATE-compressed string	MQProduce	send message to Kafka topic
Pyaes	cascading AES128 encryption/decryption	MQConsume	receive message from Kafka topic
RegExSearch	find all regular expr. matches in input	Fwrite	write to a file on the server.
RegExMatch	determine if input matches regular expr.		

*Adapted from or inspired by *FunctionBench* [4].

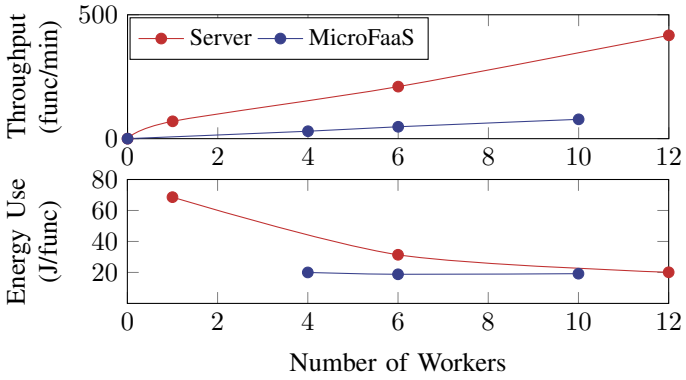

Figure 2: Throughput and energy use of *MicroFaaS on OpenFaaS* and conventional server.

Our results also show that conventional servers draw at least 4 times as much power as *MicroFaaS on OpenFaaS* to run the same function (Figure 4).

However, Figure 3 shows that *MicroFaaS on OpenFaaS* spends more time executing workload functions than the x86-based servers. Similarly, Figure 2 indicates that *MicroFaaS on OpenFaaS* has a lower throughput than that of the x86-based servers. This difference is expected due to the higher FLOPS capacity of the x86 server's processor and network card, enabling faster execution of computation- and network-intensive tasks than *MicroFaaS*.

An additional advantage of *MicroFaaS on OpenFaaS* over conventional rack servers is in cost. We conduct a cost analysis based on the total cost of ownership (TCO) model used in our previous paper [3]. Table II shows that *MicroFaaS on OpenFaaS* is ∼33% less expensive than the conventional rack server. This is mainly due to low SBC hardware costs and the energy-efficient design of *MicroFaaS*.

Comparing both iterations of *MicroFaaS*, adding OpenFaaS typically increases execution times and subsequently reduces throughput on average. However, the power consumption of both MicroFaaS iterations remain unchanged as they share the same hardware and execution model. Note that the purpose of including OpenFaaS is not to improve performance, but to provide a more accessible framework to use *MicroFaaS*.

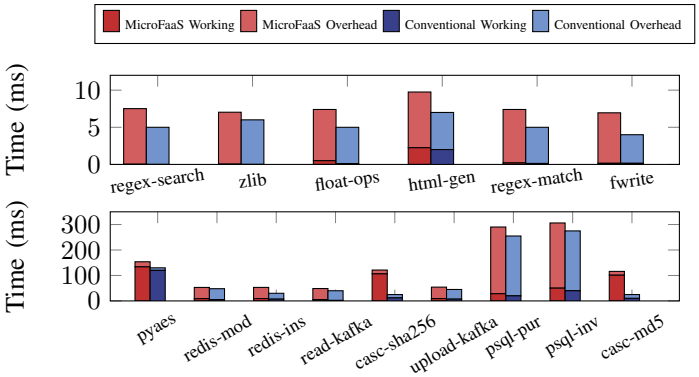

Figure 3: Performance Evaluation

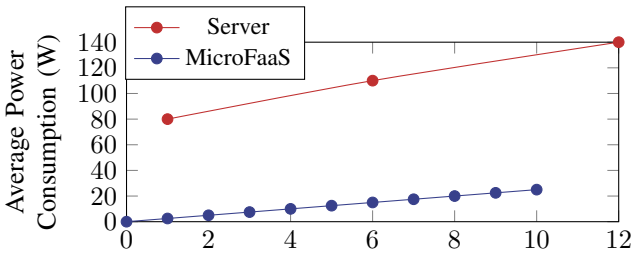

Figure 4: Power Evaluation

TABLE II: 5-Year Single Rack Lifetime Cost Comparison*

	Ideal (100% Util., 100% OR)		*Realistic (50% Util., 95% OR)*	
Expense	**Conventional**	***MicroFaaS***	**Conventional**	***MicroFaaS***
Compute	$82,451	$51,923	$86,791	$54,655
Network	$574	$12,280	$574	$12,280
Energy	$41,303	$17,884	$29,056	$11,778
Total	**$124,328**	**$82,087**	**$116,421**	**$78,713**

*Based on TCO model by Cui [5]. Costs shown in U.S. dollars.

IV. Conclusion

Previously, *MicroFaaS* demonstrated a proof-of-concept prototype that provided an energy-efficient alternative to conventional FaaS platforms. In this paper, we present *MicroFaaS on OpenFaaS* creating an easier-to-use and more accessible method of running FaaS functions on *MicroFaaS*, while maintaining energy- and cost-efficiency.

Acknowledgement

The authors thank A. Zou and Y. Pang for their contributions to the first iteration of MicroFaaS. This work has been partially funded by the Boston University UROP.

References

[1] E. Masanet *et al.*, "Recalibrating global data center energy-use estimates," *Science*, vol. 367, no. 6481, pp. 984–986, Feb. 2020. [Online]. Available: https://doi.org/ggm3sk

[2] C. Jiang *et al.*, "Energy proportional servers: Where are we in 2016?" in *37th Int. Conf. Distrib. Comput. Sys. (ICDCS)*. IEEE, 2017, pp. 1649–1660. [Online]. Available: https://doi.org/gmsk69

[3] A. Byrne *et al.*, "Microfaas: Energy-efficient serverless on bare-metal single-board computers," in *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2022, pp. 754–759.

[4] J. Kim and K. Lee, "FunctionBench: A suite of workloads for serverless cloud function service," in *12th Int. Conf. Cloud Comput. (CLOUD)*. IEEE, Jul. 2019, pp. 502–504. [Online]. Available: https://doi.org/gmsk64

[5] Y. Cui *et al.*, "Total cost of ownership model for data center technology evaluation," in *16th Intersoc. Conf. Thermal and Thermomech. Phenomena in Electron. Sys. (ITherm)*, 2017, pp. 936–942. [Online]. Available: https://doi.org/gmtnbh