



Otto-Friedrich-Universität Bamberg
Lehrstuhl für Praktische Informatik



Proposal Seminar paper

im Studiengang International Software Systems Science
der Fakultät für Wirtschaftsinformatik und Angewandte Informatik
der Otto-Friedrich-Universität Bamberg

Zum Thema:

FaaS Performance Study Methodologies

Vorgelegt von:

Hazar Gul Nazari

Themensteller:

Prof. Dr. Guido Wirtz

Abgabedatum:

17.05.2020

Contents

1	Introduction and Problem Statement	1
1.1	Introduction	1
1.2	Problem Statement	1
2	Objective and Research Questions	2
2.1	Objection	2
2.2	Research Questions	2
3	Methodology Approach	2
4	Preliminary Outline	3
5	References	4
	References	4

1 Introduction and Problem Statement

1.1 Introduction

Serverless Computing (also known as Function as a Service) is emerging as a new and compelling paradigm for the deployment of cloud applications, largely due to the recent shift of enterprise application architectures to containers and microservices [CIMS17]. According to a report published by Datadog ¹ shows nearly half of the organizations using the company's IT monitoring platform have embraced the AWS Lambda serverless computing framework. It shows companies are migrating their services toward FaaS due to its unique nature. Cloud Service Provider (CSP) companies in general provide their services in the form of "as-a-Service" and FaaS (Function as a Service) is one form of serverless cloud computing. FaaS is a category of cloud services that provides a platform allowing customers to develop, run, and manage application functionalities without the complexity of building and maintaining the infrastructure typically associated with developing and launching an app.[Rob18] In other words, FaaS is about running your backend code without the overhead of having a server to manage or you have a long-lived server which you just need to think about your application rather than server management [Rob18]. In FaaS, you are not bound to a specific language or framework, rather you can write functions on available languages or customized runtime environments. For example, AWS Lambda provides different language supports, such as NodeJS, Python, Ruby, Java, Go, .Net or custom runtime.² To deploy your code on FaaS, we need to upload the snippets of source code and everything else will be managed by the providers such as initializing virtual machines, process management, etc [Rob18].

1.2 Problem Statement

Cloud service providers deliver their products in the form of "as-a-Service", which are typically categorized by the level of abstraction [PIT19]. This approach hides the implementation details and shows only functionality to the user. However, the problem is that it is hard to measure the performance of Cloud services because they behave like black boxes. Especially with Function-as-a-Service, it is even more difficult because it completely hides server and infrastructure management from the user by design. In FaaS, there are some notable points that make the FaaS performance unpredictable, such as Cold start (bring up a new container instance when there are no active containers available for the request) [LSAP⁺18], the execution time for hardware heterogeneity [FGZ⁺18], and complex triggering mechanisms are the main factors in delaying FaaS performance [CRV10]. In the paper [PIT19], over 112 cases related to FaaS performance methodologies are reviewed which their work heavily focuses on AWS Lambda. They evaluated and studied from industry and academic literature to review different methodologies for FaaS performance benchmarks. Their evaluations are both micro-level which focuses on CPU level performance to application-level which evaluates the general performance of the application. In this paper, I will review the publications related to the FaaS performance benchmarks, summarize the important methodologies in FaaS performance evaluation and answer the questions related to FaaS performance benchmarks.

¹<https://devops.com/datadog-sees-spike-in-aws-lambda-serverless-adoption>

²<https://docs.aws.amazon.com/lambda/latest/dg/lambda-runtimes.html>

2 Objective and Research Questions

2.1 Objection

The main objective of this research is to study and review the literature on FaaS performance benchmarks and methodologies, then to summarize and present those different methodologies, frameworks, or tools that are used more to evaluate FaaS performance on different platforms. Since AWS Lambda is the leading cloud provider in the market [SL20], this paper will mainly focus on AWS Lambda and to compare different methodologies for FaaS/Serverless performance evaluation.

2.2 Research Questions

In the continuation of the above discussion, I aim to collect and review different benchmarking methodologies for FaaS performance evaluation from different resources and to find the answers to the following questions:

Research Question 1: What types of benchmark characteristics are used for FaaS/Serverless performance evaluation? and what are the differences among them?

Research Question 2: Which platforms, frameworks, and tools are commonly studied and used for FaaS/Serverless performance evaluation?

Research Question 4: How reproducible are the experiments of performance evaluation?

3 Methodology Approach

To understand what types of performance characteristics have been used for FaaS performance benchmarks, the paper [SL20] has divided the performance measurement into two different levels, the micro-benchmark which focuses on CPU and network-level performance, and application-benchmarks which evaluate the overall performance of the application, as well as some other general performance characteristics such as the use of concurrent execution, an inspection of infrastructure. With these different benchmark characteristics, I will find the answer for *Research Questions 1*.

There are two different types of platforms for FaaS configurations (private cloud and public cloud) [VEIST17]. Through OpenFaaS, you can deploy and run FaaS on private cloud³ and all well-known public cloud providers offer FaaS platforms, such as AWS Lambda, Microsoft Azure, IBM Cloud Functions, and Google Cloud Functions. the paper [SL20] extracted a list of well-known cloud providers with their supporting language runtimes and configurations which through it, I will answer the *Research question 3*.

To describe and evaluate the maturity of literature concerning reproducibility, the paper [SL20] has evaluated over 112 cases and they applied different principles and methodologies to reproduce their FaaS performance evaluation tests, such as Repeated Experiments,

³<https://www.openfaas.com/blog/ofc-private-cloud/>

Experiencement Setup Description, Probabilistic result description, Statistics evaluation, etc. Using all these steps, I will answer the *Research Question 3*.

4 Preliminary Outline

1. Introduction
2. Conceptual Foundations of FaaS
 - 2.1 Definition, Architecture and Application Area
 - 2.2 Benifits, Challenges, Requirements, and Necessity
 - 2.3 Characteristics and standards of a good benchmark
3. Related Work
 - 3.1 Characteristics of Cloud Services
 - 3.2 Benchmarking features of Serverless
4. Performance Evaluation of FaaS
 - 4.1 Publication Trends of FaaS
 - 4.2 Studied Platforms
 - 4.3 Benchmarks
 - 4.3.1 Micro-level Benchmarks
 - 4.3.2 Application-level Benchmarks
 - 4.3.2 General Charactersitics
 - 4.4 Platform Configurations
 - 4.4.1 Language Runtimes
 - 4.4.2 Function Triggers
 - 4.4.3 External Services
 - 4.5 Reproducibility
5. Conclusion

5 References

References

- [CIMS17] Paul Castro, Vatche Ishakian, Vinod Muthusamy, and Aleksander Slominski. Serverless programming (function as a service). pages 2658–2659, 06 2017.
- [CRV10] Fangzhe Chang, Jennifer Ren, and Ramesh Viswanathan. Optimal resource allocation in clouds. In *2010 IEEE 3rd International Conference on Cloud Computing*, pages 418–425. IEEE, 2010.
- [FGZ⁺18] Kamil Figiela, Adam Gajek, Adam Zima, Beata Obrok, and Maciej Malawski. Performance evaluation of heterogeneous cloud functions. *Concurrency and Computation: Practice and Experience*, 30(23):e4792, 2018.
- [LSAP⁺18] Pedro García López, Marc Sánchez-Artigas, Gerard París, Daniel Barcelona Pons, Álvaro Ruiz Ollobarren, and David Arroyo Pinto. Comparison of faas orchestration systems. In *2018 IEEE/ACM International Conference on Utility and Cloud Computing Companion (UCC Companion)*, pages 148–153. IEEE, 2018.
- [PIT19] Roland Pellegrini, Igor Ivkic, and Markus Tauber. Function-as-a-service benchmarking framework. *arXiv preprint arXiv:1905.11707*, 2019.
- [Rob18] Mike Roberts. Serverless architectures. <https://martinfowler.com/articles/serverless.html#unpacking-faas>, May 2018.
- [SL20] Joel Scheuner and Philipp Leitner. The state of research on function-as-a-service performance evaluation: A multivocal literature review. *arXiv preprint arXiv:2004.03276*, 2020.
- [VEIST17] Erwin Van Eyk, Alexandru Iosup, Simon Seif, and Markus Thömmes. The spec cloud group’s research vision on faas and serverless architectures. In *Proceedings of the 2nd International Workshop on Serverless Computing*, pages 1–4, 2017.