



Otto-Friedrich-Universität Bamberg
Lehrstuhl für Praktische Informatik



Proposal Seminar paper|Bachelor Thesis|Master Thesis

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

Zum Thema:

Thesis Topic

Vorgelegt von:

Hazar Gul Nazari

Themensteller:

Prof. Dr. Guido Wirtz

Abgabedatum:

27.05.2020

Contents

| | | |
|----------|---|----------|
| 1 | Introduction and Problem Statement | 1 |
| 2 | Conceptual Foundations | 2 |
| 3 | Objective and Research Questions | 3 |
| 4 | Methodology Approach | 4 |
| 5 | Preliminary Outline | 5 |
| | References | 6 |

1 Introduction and Problem Statement

According to a report published by Datadog¹ shows nearly half of organizations using the company's IT monitoring platform have embraced the AWS Lambda serverless computer 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.² In other words, FaaS is about running your backend code without overhead of having a server to manage or you have long-lived server which you just need to think about your application rather than server management. In FaaS, you are not bound to a specific language or framework, rather you can write functions on available languages or customized runtime environment. For example, AWS Lambda provides different language supports, such as NodeJS, Python, Ruby, Java, Go, .Net or custom runtime.³ 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. Cloud service providers deliver their products in form of *as-a-Service*; which are typically categorized by the level of abstraction. 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. Cloud service providers usually restrict the maximum size of code, memory and runtime of Cloud Functions. Nevertheless, users need clarification if more resources are needed to deliver services in high quality. Cold start, hardware heterogeneity and complex triggering mechanisms are the main factors in delaying FaaS performance. In the paper [SL20], 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. Their evaluations are both micro-level which focuses on CPU level performance to macro-level which evaluate application level performance. In this paper, I will study the research related to the FaaS performance benchmarks, summarize the important methodologies in FaaS performance evaluation and answer the questions related to FaaS performance benchmarks.

2 Conceptual Foundations

3 Objective and Research Questions

OBJECTION The main objective in this research is to study and review the previous researches on FaaS performance benchmarks and methodologies and since `scheuner2020state` has covered over 112 studies, it's more interesting to check their work, also to review the [Hup09] for performance benchmark related questions, then to summarize and present the different FaaS performance benchmarks methodologies, frameworks or tools. Since AWS Lambda is the leading cloud provider in market 5, this paper will mainly focus on AWS Lambda and to compare different methodologies for performance evaluation

RESEARCH QUESTIONS In the continue of the above discussion, my work aims to collect and review different benchmark methodology for FaaS performance benchmarks from different resources and to find answer to the following questions:

- Which tools and frameworks already exist for FaaS performance evaluation?
- Which techniques and tools are commonly studied and used?
- Which characteristics have been benchmarked for which platform?
- Difference of micro and application level benchmarks and which are typically used?
- Which micro-benchmarks are used frequently?
- Which performance characteristics related to the nature of FaaS are generally evaluated such as platform overhead and cold start?
- Which platform configurations are commonly used according to language runtime, function triggers and external services?
- How reproducible are the experiments of performance evaluation?

4 Methodology Approach

After gather The two provided sources on the VC about FaaS performance methodologies will be my main focus, specially the [SL20], since they have covered and reviewed over 112 cases (both micro-benchmark and application level benchmark analysis) of FaaS and their studies are very near up to date. As well as, I will look to the latest updates in open source frameworks in regard to FaaS benchmark frameworks. After reviewing and collecting all the found methodologies, I will summarize the overall process, compare the methods and extract the main point out of them, and present them with proper illustration. There are two levels of analysis, I will first look at micro-benchmark level analysis which is focused on the CPU level performance of FaaS in AWS and then to application level analysis which is about the overall performance of FaaS.

5 Preliminary Outline

References