Demo Abstract: A Distributed Analysis and Benchmarking Framework for Apache OpenWhisk Serverless Platform

Aleksandr Kuntsevich
Technical University of Munich
a.kuntsevich@tum.de

Pezhman Nasirifard
Technical University of Munich
p.nasirifard@tum.de

Hans-Arno Jacobsen
Technical University of Munich

ABSTRACT

Serverless computing simplifies the life cycle of scalable web applications, through delegating most of the operational concerns to the cloud providers. One prominent serverless platform is Apache OpenWhisk which is employed by IBM Cloud. Despite the apparent benefits of serverless computing, some limitations of the serverless platform, such as the state-less nature of serverless functions, can introduce scalability bottlenecks. In this work, we propose an analysis and benchmarking approach for investigating potential bottlenecks and limitations of Apache OpenWhisk serverless platform.

CCS CONCEPTS

• **Computer systems organization → Cloud computing**;

KEYWORDS

Serverless, Apache OpenWhisk, Function as a service (FaaS), Benchmarking

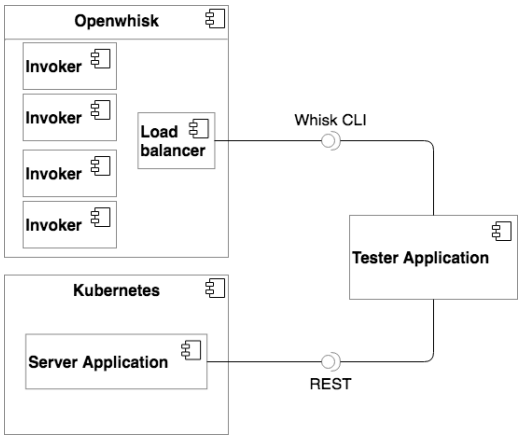

Figure 1: Deployment diagram of the framework.

1 INTRODUCTION

The serverless computing, also known as Function-as-a-Service (FaaS), represents a programming model where the developers decompose the applications into microservices, known as functions, and deploy the functions on the cloud platforms. The cloud providers are responsible for executing the functions in response to events as well as the automatic maintenance and scaling of resources. Apache OpenWhisk[1] is a production-grade open source serverless platform which is the base of IBM Cloud Functions. Although serverless paradigm offers several advantages for developing scalable distributed applications, the limitations, such as the short-lived stateless nature of the serverless functions, can cause scalability bottlenecks. To compensate for the lack of state, integrating other cloud resources, such as Database-as-a-Services is a practiced solution, but the scalability of external resources affects the scalability of the serverless application [4]. Furthermore, the automatic scaling offered by OpenWhisk is not predictable by the user, which can cause latency bottlenecks. For example, when the function is scaled down, the *cold start* problem causes latency issues [1]. In this work, we propose a benchmarking and analysis framework for executing various types of tests for investigating the potential OpenWhisk bottlenecks.

2 RELATED WORK

A few studies proposed benchmarking and analysis approaches for different serverless platforms [2, 3]. [3] developed a general cloud function benchmarking framework with CPU-intensive benchmarks for comparing the performance of several serverless platforms, including Apache OpenWhisk. In this work, we propose a benchmarking framework designed explicitly for OpenWhisk and for investigating its potential bottlenecks. Furthermore, SPEC-serverless [2] is purposed by IBM with the objective of creating standard tests for defining the baseline performance of different serverless platforms, but they have not published the benchmarks yet to the best of our knowledge.

3 TECHNICAL APPROACH

We propose several test functions for various serverless use cases. We developed an analytical framework for executing the tests on a custom installation of OpenWhisk and also a server-based application. Furthermore, the system gathers the test metrics. Figure 1 displays the deployment diagram.

3.1 OpenWhisk Set Up

To provide a testing environment that is not restricted by the cloud provider's constraints, we instrumented a distributed installation of an instance of OpenWhisk on a private cloud provided by our

[1] https://openwhisk.apache.org/

Middleware '18, December 10–14, 2018, Rennes, France

ACM ISBN 978-1-4503-6109-5/18/12…$15.00
https://doi.org/10.1145/3284014.3284016

institute. We employed four Ubuntu 16.04 virtual machines with 1 VCPU, 2.4 GB of RAM and one VM with 4 VCPUs, 9.8 GB RAM. One of the essential components of OpenWhisk is *Invoker*, which is responsible for handling the requests for function invocations. OpenWhisk also makes use of an internal load balancer for distributing the requests among the four installed invokers, as Figure 1 depicts.

3.2 Server Based Application

To examine the performance of OpenWhisk in comparison to a standard web application, and to investigate when traditional deployment models are more reasonable, we developed a simple server application for executing test functions identical to the tests on the OpenWhisk. The deployed server application is implemented in Spring Boot Framework[2] and provides a REST API. We deployed the application on a Kubernetes cluster consisting of five Ubuntu 16.04 virtual machines with allocated resources identical to the VMs used for hosting the OpenWhisk.

3.3 Tester Application

We developed a stand-alone tester application for issuing requests to the OpenWhisk and the server application. We deployed the tester application across four VMs, where each VM is configured with an OpenWhisk CLI and its OpenWhisk user, that allows the tester to perform different test scenarios independently and in parallel. We define the following parameters for testing scenarios: the number of concurrent requests sent from each tester application, the frequency of the issued requests, and the input of the function to define how resource intensive the execution is.

3.4 Test Functions

We define the following test function for executing on the platform:

- Calculation of the Nth prime number, given the N as the input. Provided the N is large enough, the calculation is CPU intensive.
- Computation of the matrix multiplication, which is RAM and CPU intensive.
- Requesting an external web resource, which shows potential I/O bottlenecks. E.g., for scenarios when IoT devices require some resources.
- Performing a query on a separate database for demonstrating another potential I/O bottlenecks.

3.5 Gathering Test Metrics

The test functions are implemented in JavaScript and Java, and the tester applications execute the functions according to the test scenarios. Furthermore, system metrics are monitored, including the CPU, RAM, disk usage and the latency for executing function invocations. The system metrics measure the latency of function invocations, concerning the cold start problem, and how the system scales up and down. As an example, Figure 2 presents the latency for the increasing number of requests of the OpenWhisk functions written in Java and JavaScript in comparison to the Spring web-based application executing the same function. In this scenario, we used the test function for calculating the Nth prime number.

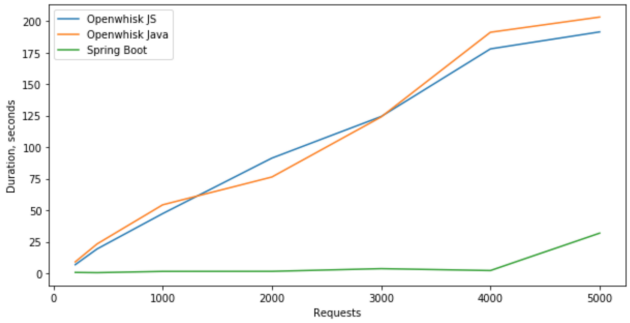

Figure 2: Executing the calculation of Nth prime number test scenario.

We open-sourced the system, and the source code is publicly available[3].

4 CONCLUSIONS

In this demo, we proposed a technical solution for benchmarking and analysis of Apache OpenWhisk platform by using a set of test functions. This framework provides an opportunity to investigate potential bottlenecks and limitations of OpenWhisk.

ACKNOWLEDGMENTS

Alexander von Humboldt Foundation supported this project.

REFERENCES

[1] I. Baldini, P. C. Castro, K. Chang, P. Cheng, S. J. Fink, V. Ishakian, N. Mitchell, V. Muthusamy, R. M. Rabbah, A. Slominski, and P. Suter. 2017. *Serverless Computing: Current Trends and Open Problems.* Springer Singapore.

[2] N. Kaviani and M. Maximilien. 2018. CF Serverless: Attempts at a Benchmark for Serverless Computing. https://docs.google.com/document/d/1e7xTz1P9aPpb0CFZucAAI16Rzef7PWSPLN71pNDa5jg/edit. Accessed: 2018-08-16.

[3] M. Malawski, K. Figiela, A. Gajek, and A. Zima. 2018. Benchmarking Heterogeneous Cloud Functions. In *Euro-Par 2017: Parallel Processing Workshops.* Springer International Publishing.

[4] P. Nasirifard, A. Slominski, V. Muthusamy, V. Ishakian, and H. A. Jacobsen. 2017. A Serverless Topic-based and Content-based Pub/Sub Broker: Demo. In *Proceedings of the 18th ACM/IFIP/USENIX Middleware Conference: Posters and Demos.* ACM.

[2] https://spring.io/projects/spring-boot

[3] https://github.com/i13tum/openwhisk-bench