universität innsbruck

High-Level Modeling and Low-Level Adaption of Serverless Function Choreographies

Benjamin Walch

Bachelor Thesis

Supervisor:
Dr. Shashko Ristov
Department of Computer Science
Universität Innsbruck

Innsbruck, December 12, 2019

Eidesstaatliche Erklärung

Ich erkläre hiermit an Eides statt durch meine eigenhändige Unterschrift, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe. Alle Stellen, die wörtlich oder inhaltlich den angegebenen Quellen entnommen wurden, sind als solche kenntlich gemacht.

Ich erkläre mich mit der Archivierung der vorliegenden Bachelorarbeit einverstanden.

Datum	Unterschrift

Abstract

"Run code, not Server" is the most recent term of cloud computing providers. With the rise of the serverless technology during the last years, *FaaS* became more and more popular. The Distributed and Parallel Systems Group from University of Innsbruck are doing research in this topic. One of the results of this research is an API, which was developed for describing serverless application workflows programmatically. The product which results in using that API is the workflow being described in a generated YAML file. This file can be further processed by (other) machines.

The aim of this bachelor project is to develop a visual workflow editor, which makes modeling of workflows possible at a high level of abstraction. Additionally, composed workflows can be saved, reopened and edited. The tool should also be able to optimize given workflows for multiple *FaaS* provider(s) in case of quotas and limits, and also in case of performance.

Contents

1 Introduction

With the rise of *FaaS*, a high level of flexibility in execution of code appeared. Global Players like Amazon, Google, IBM and Microsoft jumped on the train and provide their infrastructure to Developers, able to deploy functions and execute them in the cloud. Each system has its own definitions on how to define, deploy, run and execute code.

1.1 Motivation

2 Background

2.1 FaaS

2.2 AFCL

3 Overview

4 Software Architecture

4.1 Frontend

4.2 Backend

5 Outlook

6 Conclusion