Design and Implementation of Distributed Application: Project Report

Wallace Garbim
Instituto Superior Tecnico

Miguel Crespo
Instituto Superior Tecnico

Florian Ehrenstorfer
Instituto Superior Tecnico
fehrenstorfer@gmail.com

Abstract

In this paper we present a simplified function-as-a-service cloud platform, which enables the distributed execution of a chain of operators on a shared storage. The system consists of one scheduler and multiple worker and storage nodes. Workload is balanced equally between workers through the scheduler. Data consistency and fault tolerance are guaranteed by the storage nodes. We present the problem and motivation first, then describe our solution in detail, and evaluate it at last in this paper.

1. Introduction

Distributed function-as-a-service cloud platforms are increasing in popularity with big cloud providers like AWS and Google Cloud offering custom solutions. They offer great flexibility with of the shelf plug and play options as well as customizable and self-developed options. For this project we developed our own distributed function-as-a-service cloud platform, a simplified version of other platforms. We focused on its design, implementation and evaluation in the project in this paper.

2. Problem

The function-as-a-service cloud platform allows to run applications in a distributed network. Applications consist of a chain of operators, which can access a shared storage system. Because the storage system is distributed as well, one of the main problems is data consistency and fault tolerance. In order to maintain data consistency a gossip protocol is used to propagate updates through the network. The storage system supports three operations: read, write, and updateIfValueIs. Records are stored with the value and a version number. For the identification of the correct storage server to contact about a value, consistent hashing is used. For data consistency, operators have to read consistent version of the data, which has to be ensured by the system. Because the system also has to provide fault-tolerance, data has to be distributed between storage nodes, and reads have to be redirected. The different operators are assigned to workers by a scheduler, and then send from one worker to another after the operator has finished.

3. Solution

3.1. Design

3.2. Implementation

4. Evaluation

5. Conclusion