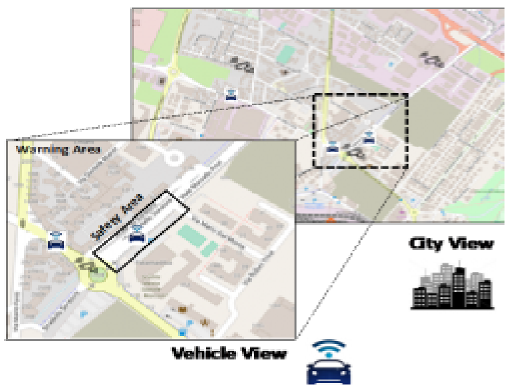
Contenuto archiviato il 2023-04-12

Using OpenWhisk as a Polyglot Real-Time Event-Driven Programming Model in CLASS

The CLASS project aims to address smart city challenges by facilitating efficient development and execution of data processing from edge to cloud resources and enabling real-time guarantees. A key enabler of these capabilities is real-time event handling, which is planned to be delivered using a serverless platform of Apache OpenWhisk, adding powerful properties of polyglot programming and easy to use, dynamic configuration.



© CLASS Project

One of the more interesting aspects of the CLASS project is taking computation out to the real world. While focusing on use cases of smart cities and smart traffic, CLASS aims to provide a generic platform for processing events with computation that spans hardware at the edge, IoT devices in the field, and the cloud. The best way of processing each event may vary – it may involve big data analytics requiring specific models, Complex Event Processing (CEP), a simple sequential code, etc. In other words, a developer may need to be able to choose to find the right tool for the job. On top of that, real-world events often need to be processed in real time, obeying a predefined deadline or rate of computation.

Take for example the CLASS use case of obstacle detection. The idea is to alert a car's driver of possible collisions with other objects within the car's warning area, which is an area formed around the car's location by city sensors (mostly cameras) and extends well beyond the car's potential sensor coverage (safety area) and surroundings such as buildings. This extends the driver's awareness to potential collisions when turning corners, hitting pedestrians walking around other cars, etc.

Identifying the collisions may involve handling multiple streams of events. For example, there will be one event stream from each smart camera, providing updates of detected objects. There will be another event stream from each car, identifying its current location, speed and direction. Then there may be timer events to compute the content of each car's warning area based on its recent location and recent data from cameras. The results can be fed into another engine to compute trajectories and identify possible collisions, which can be sent in turn as events to the driver's smart console in the car to generate alerts.

Furthermore, the response to each such event may be handled differently. Updates from sources can be written as sequential code and distributed using COMPSs. Filtering the objects inside a warning area is a good fit for a map/reduce engine, e.g. PyWren. Trajectories and potential collisions can be inferred using a CEP engine. Each of these platforms may require using specific programming languages, so the overall event handling fabric needs to be, in fact, polyglot.

So how can this be accomplished? Enter Apache OpenWhisk – OW for short. It's a serverless platform, which means developers deploy code directly into it (like PaaS clouds) without specifying server or deployment location. The basic unit of code is a function[i], also called an action in OW. It's a sequential piece of code that accepts parameters, does some computation and returns a result. In OW, actions can be built from any piece of executable code in any language, so it's purely polyglot. Once an action is created, it can be invoked over OW's REST interface.

In the context of the CLASS project, we are working on extending OW with real-time support, in three complementary situations: one is monitoring and enforcement of deadlines, from event trigger to action completion. The second is a calibration mechanism for developer code, so that with high probability it should complete within a predefined deadline. The third, which is joint work across the project, involves employing schedulers in the cloud and at the edge that will help guarantee predictable computation. In addition, we are extending PyWren on OpenWhisk to demonstrate a first-of-a-kind support for serverless map/reduce computation with real-time constraints.

Parole chiave

CLASS

Big data

Connected cars

Paesi

Spain

Contributore

Contributo di

BSC

Spagna

Sito web

Progetti correlati

HORIZON 2020

PROGETTO

Edge and CLoud Computation: A Highly Distributed Software Architecture for Big Data AnalyticS

CLASS

13 Settembre 2023

Ultimo aggiornamento: 27 Novembre 2018

Permalink: https://cordis.europa.eu/article/id/124224-using-openwhisk-as-a-polyglot-realtime-eventdriven-programming-model-in-class/it