Informatics Institute of Technology

In collaboration with

University of Westminster, UK

Cloud-native Dynamic Scalability for Distributed Stream Processing Systems

A Dissertation By

Senthuran Ambalavanar

2015215 | w1608452

Supervised By

Mr. Saman Hettiarachchi

Submitted in partial fulfilment of the requirements for the

BEng (Hons) Software Engineering degree

Department of Computing

May 2019

Abstract

Distributed stream processing systems separate streaming applications into executable smaller units known as child streaming applications, distribute them across several nodes in clusters and run, in order to cater heavy and growing stream processing use cases. Due to the tremendous benefits provided by cloud infrastructures, distributed stream processing systems are often adopted to be deployed on cloud environments. Scalability is one of the common goals addressed by such systems. With varying amount of resource and event consumptions during the time of execution, the ability of individually scaling each child streaming application as required, increases the efficiency of scalability.

Some child streaming applications frequently communicate with their required resources, such as a MySQL database. In cloud environments, network overheads are possible, when the resource and the consuming child streaming application are located in different nodes within the cluster.

Focusing on scaling stateless operators in Kubernetes environment, this project addresses these concerns through a framework, for stream processors that offer text-based streaming application development.

By natively using Kubernetes container orchestrator to create dedicated Workers for executing child streaming applications, the framework addresses the concern of individually scaling each child streaming application that has different event consumption rates, as required.

Collocation of a special resource and its consumer in the same node of the cluster, will reduce possible network overheads, which can facilitate scalability. This has been achieved with Kubernetes pod affinity. A resource requirement for a child streaming application is identified at the time of parsing it, and the containing Worker is applied with labels in order to be eligible for pod affinity based scheduling. By natively using Kubernetes for resource management, the project improves resource utilisation and latency of distributed stream processing systems in cloud environments, and produces a reliable and easy to deploy solution.

Keywords:

Cloud computing, Distributed Stream Processing, Container Orchestrator, Dynamic Scaling, Dynamic Scheduling, Streaming Application