Accepted Manuscript

Decentralized self-adaptation for elastic Data Stream Processing

Valeria Cardellini, Francesco Lo Presti, Matteo Nardelli, Gabriele Russo Russo

PII: S0167-739X(17)32682-1
DOI: https://doi.org/10.1016/j.future.2018.05.025
Reference: FUTURE 4200

To appear in: *Future Generation Computer Systems*

Received date : 17 November 2017
Revised date : 7 April 2018
Accepted date : 11 May 2018

Please cite this article as: V. Cardellini, F.L. Presti, M. Nardelli, G.R. Russo, Decentralized self-adaptation for elastic Data Stream Processing, *Future Generation Computer Systems* (2018), https://doi.org/10.1016/j.future.2018.05.025

Decentralized Self-Adaptation for Elastic Data Stream Processing

Valeria Cardellini[a], Francesco Lo Presti[a], Matteo Nardelli[a,*], Gabriele Russo Russo[a]

[a]*Department of Civil Engineering and Computer Science Engineering*
University of Rome Tor Vergata, Italy

Abstract

Data Stream Processing (DSP) applications are widely used to develop new pervasive services, which require to seamlessly process huge amounts of data in a near real-time fashion. To keep up with the high volume of daily produced data, these applications need to dynamically scale their execution on multiple computing nodes, so to process the incoming data flow in parallel.

In this paper, we present a hierarchical distributed architecture for the autonomous control of elastic DSP applications. It consists of a two-layered hierarchical solution, where a centralized per-application component coordinates the run-time adaptation of subordinated distributed components, which, in turn, locally control the adaptation of single DSP operators. Thanks to its features, the proposed solution can efficiently run in large-scale Fog computing environments. Exploiting this framework, we design several distributed self-adaptation policies, including a popular threshold-based approach and two reinforcement learning solutions. We integrate the hierarchical architecture and the devised self-adaptation policies in Apache Storm, a popular open-source DSP framework. Relying on the DEBS 2015 Grand Challenge as a benchmark application, we show the benefits of the presented self-adaptation policies, and discuss the strengths of reinforcement learning based approaches, which autonomously learn from experience how to optimize the application performance.

Keywords: Data Stream Processing, Self Adaptive, Hierarchical Control, MAPE, Reinforcement Learning

1. Introduction

The ubiquitous presence of sensing devices, which continuously produce streams of data, creates a fertile ground for the development of new and pervasive services. An efficient use of those data can improve the quality of everyday life in many cross-concern domains, including health-care, energy management, logistic, and transportation. Data Stream Processing (DSP) represents a prominent approach to elaborate data as soon as they are generated, thus enabling the design of near real-time applications.

A DSP application is represented as a directed acyclic graph (DAG), with data sources, operators, and final consumers as vertices, and streams as edges. Each *operator* can be seen as a black-box processing element that continuously receives incoming streams, applies a transformation, and generates new outgoing streams. In modern scenarios, DSP applications are characterized by strict latency requirements in face of variable and high data volumes to process. To deal with operator overloading, a commonly adopted stream processing optimization is *data parallelism*, which enables to process data in parallel on multiple computing nodes (given that a single machine cannot provide enough processing power). Data parallelism consists in scaling-out or scaling-in the number of parallel instances for the operators, so that each instance processes a subset of the incoming data flow (e.g., [1]). As the application workloads are typically highly variable, the application parallelism should *elastically* self-adapt at run-time to match the workload and prevent resource wastage.

Moreover, since data sources are in general geographically distributed (e.g., in IoT scenarios), recently we also have witnessed a paradigm shift with DSP applications being deployed over distributed Cloud and Fog computing resources. This computing environment *de facto* brings applications closer to the data, rather than the other way around, to reduce application latency and make better use of the ever increasing amount of resources at the network edges. Nevertheless, this very idea makes it challenging to control DSP application performance. Most of the approaches proposed in literature for the DSP application deployment and adaptation have been designed for cluster environments with a centralized control component, that can benefit from a global system view. These solutions typically do not scale well in a highly distributed environment, given the spatial distribution, heterogeneity, and sheer size of the infrastructure itself. In fact, modern DSP systems should be able to seamlessly deal with a large number of interconnected small and medium size devices (e.g., IoT devices), which continuously emit and consume data (e.g., in a smart city). To improve scalability, several decentralized management solutions have been proposed, e.g., [2, 3, 4]. Devising a decentralized policy that reconfigures the DSP application deployment exploiting only a local system view is, in general, not trivial. Indeed, the inherent lack of coordination of decentralized solu-

*Corresponding author
Email addresses: cardellini@ing.uniroma2.it (Valeria Cardellini), lopresti@info.uniroma2.it (Francesco Lo Presti), nardelli@ing.uniroma2.it (Matteo Nardelli), russo.russo@ing.uniroma2.it (Gabriele Russo Russo)