Accepted Manuscript

Elastic-PPQ: A two-level autonomic system for spatial preference query processing over dynamic data streams

Gabriele Mencagli, Massimo Torquati, Marco Danelutto

PII: S0167-739X(17)30938-X
DOI: http://dx.doi.org/10.1016/j.future.2017.09.004
Reference: FUTURE 3659

To appear in: *Future Generation Computer Systems*

Received date : 14 May 2017
Revised date : 23 July 2017
Accepted date : 3 September 2017

Please cite this article as: G. Mencagli, M. Torquati, M. Danelutto, Elastic-PPQ: A two-level autonomic system for spatial preference query processing over dynamic data streams, *Future Generation Computer Systems* (2017), http://dx.doi.org/10.1016/j.future.2017.09.004

Elastic-PPQ: a Two-level Autonomic System for Spatial Preference Query Processing over Dynamic Data Streams

Gabriele Mencagli*, Massimo Torquati and Marco Danelutto

Department of Computer Science, University of Pisa
Largo B. Pontecorvo 3, I-56127 Pisa, Italy

Abstract

Paradigms like *Internet of Things* and the most recent *Internet of Everything* are shifting the attention towards systems able to process unbounded sequences of items in the form of data streams. In the real world, data streams may be highly variable, exhibiting *burstiness* in the arrival rate and *non-stationarities* such as trends and cyclic behaviors. Furthermore, input items may be not ordered according to timestamps. This raises the complexity of stream processing systems, which must support elastic resource management and autonomic QoS control through sophisticated strategies and run-time mechanisms. In this paper we present `Elastic-PPQ`, a system for processing spatial preference queries over dynamic data streams. The key aspect of the system design is the existence of two adaptation levels handling workload variations at different time-scales. To address fast time-scale variations we design a fine regulatory mechanism of load balancing supported by a control-theoretic approach. The logic of the second adaptation level, targeting slower time-scale variations, is incorporated in a *Fuzzy Logic Controller* that makes scale in/out decisions of the system parallelism degree. The approach has been successfully evaluated under synthetic and real-world datasets.

Keywords: Data Stream Processing, Sliding-Window Queries, Elasticity, Burstiness, Out-of-order Data Streams, Intel Knights Landing

1. Introduction

In our ever-more connected world we are assisting to an unprecedented diffusion of systems and devices able to generate massive streams of transient data transmitted at great velocity. An urgent challenge is to design computing and communication infrastructures supporting real-time processing of data streams in order to extract complex analytics for decision making [1].

The *Data Stream Processing* paradigm [2] has been proposed to cope with this issue. It consists in a programming model where applications are written as data-flow graphs of logic entities called *operators*, each one applying transformations on the input data and deployed on parallel and possibly distributed environments.

Real-world data streams may exhibit dynamic characteristics like hysteric data rates with abrupt surges and out-of-order arrivals, where stream elements (called *tuples*) may not be gathered in increasing order of timestamps. Due to such erratic nature, *a-priori* capacity planning and management of the resources needed to execute streaming applications is not always practical [3], and this makes difficult to maintain the desired *Quality of Service* (QoS) while achieving high resource utilization efficiency.

To circumvent this issue, some papers have investigated the problem of enhancing stream processing systems with elasticity mechanisms [4, 5, 6, 7]. The term *elasticity* has been adopted in the field of Cloud Computing [8, 9] to indicate the ability to change the number and the size of virtual machines to adapt to the workload level. Recently, the same term has been used in the data stream processing literature [3, 4, 5] to indicate systems where the *parallelism level* of some operators (i.e. the number of replicas of the same operator working on distinct input data in parallel) can be dynamically modified to optimize the application throughput in response to variations in the arrival rate.

Most of the past elastic approaches have been experimented assuming workload variations in the form of deterministic trends and non-stationarities observed at time-scales of tens of seconds [3] or even minutes/hours [9]. The work in [10] demonstrated that in elastic Clouds the presence of fast variabilities such as *burstiness* may be detrimental both for the system QoS and for the resource utilization rate. This negative effect also occurs in stream processing systems, where phases with arrivals occurring in clusters (temporal burstiness) are likely to generate a growing computational burden to the system.

Furthermore, this problem is exacerbated by the fact that scheduling strategies may generate load imbalance in

*Corresponding author, Phone: +39-050-221-3132
Email addresses: `{mencagli, torquati, danelutto}@di.unipi.it` (Gabriele Mencagli*, Massimo Torquati and Marco Danelutto)