ARTICLE IN PRESS YJCSS:3034

Journal of Computer and System Sciences ••• (••••) •••–•••

Contents lists available at ScienceDirect

Journal of Computer and System Sciences

www.elsevier.com/locate/jcss

Elasticity management of Streaming Data Analytics Flows on clouds

Alireza Khoshkbarforoushha [a,b,*], Alireza Khosravian [c], Rajiv Ranjan [d]

[a] Research School of Computer Science, The Australian National University, Canberra ACT 0200, Australia
[b] Data61, CSIRO, Canberra ACT 0200, Australia
[c] School of Computer Science, University of Adelaide, Adelaide SA 5005, Australia
[d] School of Computing Science, Newcastle University, Newcastle upon Tyne, United Kingdom

ARTICLE INFO

Article history:
Received 15 January 2016
Received in revised form 2 November 2016
Accepted 7 November 2016
Available online xxxx

Keywords:
Data analytics flow
Control theory
Data-intensive workloads
Resource management
Public clouds
Multi-objective optimization

ABSTRACT

In this paper, we present a framework for resource management of Streaming Data Analytics Flows (SDAF). Using advanced techniques in control and optimization theory, we design an adaptive control system tailored to the data ingestion, analytics, and storage layers of the SDAF that is able to continuously detect and self-adapt to workload changes for meeting the users' service level objectives. Our experiments based on a real-world SDAF show that, the proposed control scheme is able to reduce the deviation from desired utilization of resources by up to 48% compared to existing techniques.

1. Introduction

Growing attention to getting real-time insights into streaming data leads to the formation of many complex Streaming Data Analytics Flows (SDAF). For example, by analyzing data using streaming data analytics flows, real-time situational awareness can be developed for handling events such as natural disasters, traffic congestion, or major traffic incidents [1].

An SDAF typically consists of three layers: data ingestion, analytics, and storage [2,3]. The data ingestion layer accepts data from multiple sources such as online services or back-end system logs. The data analytics layer consists of many platforms such as batch and stream processing systems. The ingestion and analytics layers make use of different databases during execution and where required persist the data in the storage layer.

1.1. Research motivation and challenges

An SDAF is formed via orchestration of different data processing platforms across a network of unlimited computing and storage resources [5]. For example, Fig. 1 shows the Amazon reference SDAF[1] that performs real-time analysis over click

* Corresponding author at: Research School of Computer Science, The Australian National University, Canberra ACT 0200, Australia.
E-mail addresses: a.khoshkbarforoushha@anu.edu.au (A. Khoshkbarforoushha), alireza.khosravian@adelaide.edu.au (A. Khosravian), raj.ranjan@ncl.ac.uk (R. Ranjan).

[1] https://github.com/awslabs/aws-big-data-blog/tree/master/aws-blog-kinesis-storm-clickstream-app.

http://dx.doi.org/10.1016/j.jcss.2016.11.002

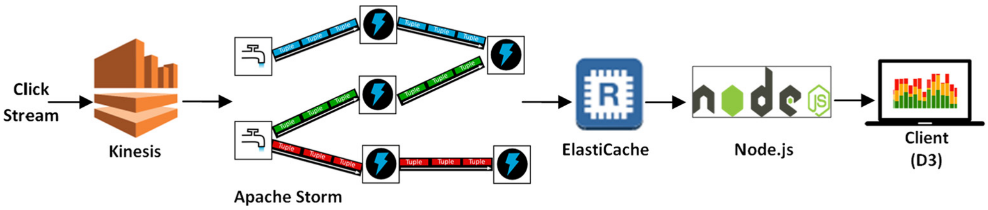

Fig. 1. A data analytics flow that performs real-time sliding-windows analysis over click stream data [4].

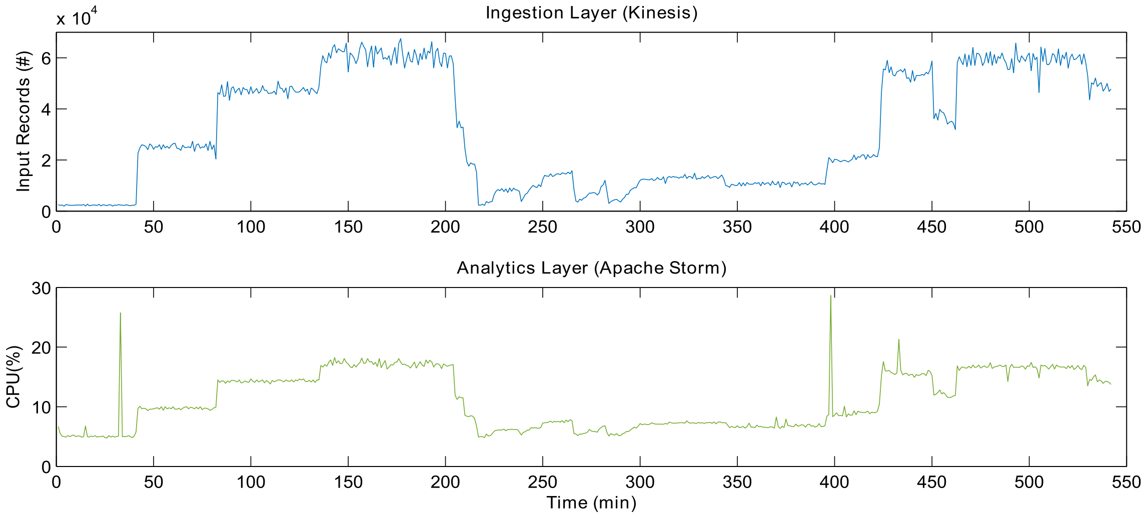

Fig. 2. The data arrival rate at the ingestion layer (Amazon Kinesis in Fig. 1) is strongly correlated (coefficient = 0.95) with the CPU load at the analytics layer (Apache Storm).

stream data. In this architecture, Kinesis acts as a distributed messaging system, Apache Storm[2] as a distributed real-time computation system, and ElastiCache[3] as a persistent storage.

Despite straightforward orchestration, elasticity management of the established flow has unique challenges because it needs to cover three aspects: i) scalability, the ability to sustain workload fluctuations, ii) cost efficiency, acquiring only the required resources, iii) time efficiency, resources should be acquired and released as soon as possible [6–8].

Recent review of elasticity techniques [9] stresses lack of a *holistic* approach in resource requirements management of workloads, whereas [10] shows that the ability to scale down both web servers and cache tier leads to 65% saving of the peak operational cost, compared to 45% if we only consider resizing the web server tier. This leads us to the first research problem: *How much resource capacity should be allocated to different big data platforms within an SDAF such that Service Level Objectives (SLO) are continuously met?* There are two challenges in response to this question:

Workload dependencies. In an SDAF, workloads pertaining to different platforms are dependent on each other. For example, Fig. 2 clearly shows how the workload dynamics in the data injection layer can be traced down to the analytics layer, since the input records in ingestion layer strongly correlated with CPU usage in the analytics layer. To provide smooth elasticity management, these dependencies need to be detected dynamically. However, existing approaches perform resource allocation across different layers irrespective of inter-layer workload dependencies which make them incapable of ensuring SLOs for emerging classes of SDAFs.

Different cloud services and monetary schemes. An SDAF is built upon multiple big data processing platforms and hardware resources offered by public clouds that adopt multiple pricing schemes [11]. For example, Amazon Kinesis's[4] pricing is based on two dimensions including shard hour[5] and PUT Payload unit, whereas ElastiCache has one hourly pricing scheme which is based on the cache node type. To meet the SLOs (e.g. budget constraint) for a data analytics flow, resource requirements and their associated cost dimensions have to be considered during the allocation process.

Once the resource shares are determined, adaptive and timely provisioning of the resources is yet another issue which forms the second key research problem of this paper: *How can we sustain resource requirements of the SDAF in a timely manner?* In this regard, we face the following challenge:

[2] http://storm.incubator.apache.org.
[3] http://aws.amazon.com/elasticache.
[4] http://aws.amazon.com/kinesis.
[5] In Kinesis, a stream is composed of one or more shards, each provides a fixed unit of capacity.

Please cite this article in press as: A. Khoshkbarforoushha et al., Elasticity management of Streaming Data Analytics Flows on clouds, J. Comput. Syst. Sci. (2016), http://dx.doi.org/10.1016/j.jcss.2016.11.002