Knowledge-Based Systems 79 (2015) 3–17

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

A knowledge-based platform for Big Data analytics based on publish/subscribe services and stream processing

Christian Esposito [a,*], Massimo Ficco [b,1], Francesco Palmieri [b,1], Aniello Castiglione [c,2]

[a] Institute of High Performance Computing and Networking (ICAR), National Research Council, Via Pietro Castellino 111, I-80131 Napoli, Italy
[b] Department of Industrial and Information Engineering, Second University of Naples, Via Roma 29, I-81031 Aversa, CE, Italy
[c] Department of Computer Science, University of Salerno, Via Giovanni Paolo II, 132, I-84084 Fisciano, SA, Italy

ARTICLE INFO

Article history:
Available online 15 May 2014

Keywords:
Publish/subscribe services
Interoperability
Schema matching
Semantic search
Complex event processing
Big Data analytics
Ontologies

ABSTRACT

Big Data analytics is considered an imperative aspect to be further improved in order to increase the operating margin of both public and private enterprises, and represents the next frontier for their innovation, competition, and productivity. Big Data are typically produced in different sectors of the above organizations, often geographically distributed throughout the world, and are characterized by a large size and variety. Therefore, there is a strong need for platforms handling larger and larger amounts of data in contexts characterized by complex event processing systems and multiple heterogeneous sources, dealing with the various issues related to efficiently disseminating, collecting and analyzing them in a fully distributed way.

In such a scenario, this work proposes a way to overcome two fundamental issues: data heterogeneity and advanced processing capabilities. We present a knowledge-based solution for Big Data analytics, which consists in applying automatic schema mapping to face with data heterogeneity, as well as ontology extraction and semantic inference to support innovative processing. Such a solution, based on the publish/subscribe paradigm, has been evaluated within the context of a simple experimental proof-of-concept in order to determine its performance and effectiveness.

1. Introduction

At the state of the art, large and complex ICT systems are designed by assuming a system of systems perspective, i.e., a large number of components integrated by means of middleware adapters/interfaces over a wide-area communication network. Such systems usually generate a large amount of loosely structured data sets, often known as Big Data, since they are characterized by a huge size and an high degree of complexity, that need to be effectively stored and processed [1,2]. Some concrete examples can be taken from the application domains of environmental monitoring, intrusion/anomaly detection systems, healthcare management and online analysis of financial data, such as stock price trends. The analysis of such data sets is becoming vital for the success of a business and for the achievement of the ICT mission for the involved organizations. Therefore, there is the need for extremely efficient and flexible data analysis platforms to manage and process such data sets, sometimes on a online/timely basis. However, their huge size and variety are limiting the applicability of the traditional data mining approaches, which typically encompass a centralized collector, able to store and process data, that can become an unacceptable performance bottleneck. Consequently, the demand for a more distributed approach involving the scalable and efficient management of Big Data is strongly increasing in the current business arena.

The well-known *MapReduce* paradigm [3] has attracted great interest, and is currently considered the winning-choice framework for large-scale data processing. Such a successful adoption both in industry and academia is motivated by its simplicity, scalability and fault-tolerance features, and further boosted by the availability of an open source implementation offered by Apache (namely Hadoop [4]). Despite such a great success and benefits, MapReduce exhibits several limitations, making it unsuitable for the overall spectrum of needs for large-scale data processing. In particular, as described in details in [5], the MapReduce paradigm is affected by several performance limitations, introducing high latency in data access and making it not suitable for interactive

* Corresponding author. Tel.: +39 081 6139508 (O); fax: +39 081 6139531.
E-mail addresses: christian.esposito@na.icar.cnr.it (C. Esposito), massimo.ficco@unina2.it (M. Ficco), francesco.palmieri@unina.it (F. Palmieri), castiglione@ieee.org, castiglione@acm.org (A. Castiglione).
[1] Tel.: +39 081 5010505 (O); fax: +39 081 5010203.
[2] Tel.: +39 089 969594 (O); fax: +39 089 969600.

http://dx.doi.org/10.1016/j.knosys.2014.05.003
0950-7051/

use. As a matter of fact, Hadoop is built on top of the Hadoop Distributed File System (HDFS), a distributed file system designed to run on commodity hardware, and more suitable for batch processing of very large amounts of data rather than for interactive applications. This makes the MapReduce paradigm unsuitable for event-based online Big Data processing architectures, and motivates the need of investigating other different paradigms and novel platforms for large-scale event stream-driven analytics solutions.

Starting from these considerations, the main aim of this work is to design and implement a flexible architectural platform providing distributed mining solution for huge amounts of unstructured data within the context of complex event processing systems, allowing the easy integration of a large number of information sources geographically scattered throughout the world. Such unstructured data sources (such as Web clickstream data, online social network activity logs, data transfer or phone calls records, and flight tracking logs) usually do not fit into more traditional data warehousing or business intelligence techniques/tools and sometimes require timely correlation/processing triggered on specific event basis (e.g., in case of online analysis solicited by specific crisis conditions or emotional patterns). This implies the introduction of new flexible integration paradigms, as well as knowledge-driven semantic inference features in data retrieval and processing to result in really effective business benefits. Publish/subscribe services [6,7] have been proved to be a suitable and robust solution for the integration of a large number of heterogeneous entities thanks to their intrinsic asynchronous communication and decoupling properties. In fact, these properties remove the need of explicitly establishing all the dependencies among the interacting entities, in order to make the resulting virtualized communication infrastructure more scalable, flexible and maintainable. In addition, despite their inherently asynchronous nature, publish/subscribe services ensure timely interactions, characterized by low-latency message delivery features, between the corresponding parties, being also perfectly suitable in online event-driven data processing systems.

Accordingly, we have designed our Big Data analytics architecture by building it on top of a publish/subscribe service stratum, serving as the communication facility used to exchange data among the involved components. Such a publish/subscribe service stratum brilliantly solves several interoperability issues due to the heterogeneity of the data to be handled in typical Big Data scenarios. In fact, most of the large-scale infrastructures that require Big Data analytics are rarely built ex-novo, but it is more probable that they are realized from the federation of already existing legacy systems, incrementally developed over the years by different companies in order to accomplish the customer needs known at the time of realization, without an organic evolution strategy. For this reason, the systems to be federated are characterized by a strong heterogeneity, that must be coped with by using abstraction mechanisms available on multiple layers [8,9]. Therefore, such systems can be easily interconnected by means of publish/subscribe services, with the help of proper adapters and interfaces in order to overcome their heterogeneity and make them fully interoperable on a timely basis. We can distinguish the aforementioned heterogeneity both at the syntactic and semantic level. That is, each system is characterized by a given schema describing the data to be exchanged. Even in domains where proper standards have been issued and progressively imposed, the heterogeneity in the data schema is still seen as a problem. Such heterogeneity limits the possibility for applications to comprehend the messages received from a different system, and hence to interoperate. Specifically, publish/subscribe services use these data schemas to serialize and deserialize the data objects to be exchanged over the network. If the schema known by the destination is different than the one applied by the source, it is not possible to correctly deserialize the arrived message, with a consequent loss of information. Interoperability not only has to resolve the differences in data structures, but it also has to deal with semantic heterogeneity. Each single value composing the data to be exchanged can have a different definition and meaning on the interacting systems. Thus, we propose a knowledge-based enforcement for publish/subscribe services in order to address their limitations in supporting syntactic and semantic interoperability among heterogeneous entities. Our driving idea is to integrate schema matching approaches in the notification service, so that publishers and subscribers can have different data schemas as well as exchange events that are easy to be understood and processed.

In order to be processed online, in a fully distributed (and hence more scalable) way, Big Data are filtered, transformed and/or aggregated along the path from the producers to the consumers, to allow consumers to retrieve only what they are interested in, and not all the data generated by the producers. This allows avoiding the performance and dependability bottlenecks introduced by a centralized collecting and processing unit, and guarantees a considerable reduction of the processing latency as well as of the traffic imposed on the communication network (since processing is placed closer to the event producers), with considerable benefits in terms of network resource usage. For this purpose, we introduced on top of the publish/subscribe service an event stream processing layer [10], which considers data as an almost continuous stream of events. This event stream is generated by several producers and reaches its destinations by passing through a series of processing agents. These agents are able to apply a series of operations taken from the available complex event processing techniques portfolio [10] to filter parts of the stream, merge two or more distinct streams, perform queries over a stream and to persistently store streams. Hence, the first step of our work consisted in the definition and implementation of several primitive stream-processing operators specialized as data processing agents and in the engineering of a model-based prototype to assist Big Data analysts to easily create a stream processing infrastructure based on publish/subscribe services.

Furthermore, we also observed that traditional solutions for performing event stream processing are affected by two main problems limiting their applicability to Big Data analytics:

- *Stream Interoperability*, i.e., users are exposed to the heterogeneity in the structures of the different event streams of interest. In fact, users have to know the details of the event types in order to properly define query strings based on the stream structures, and to write different queries for streams whose structure varies.
- *Query Expressiveness*, i.e., events composing the streams are considered as a map of attributes and values, and the typical queries on event streams are structured in order to find particular values in the events.

The construction of our platform on top of a publish/subscribe service model, empowered with a knowledge-based solution for interoperability among heterogeneous event types, allows us to easily resolve Stream Interoperability issues, leaving only the Query Expressiveness as an open problem. Recent research on event-driven systems, such as the works described in [11,12], is speculating on the introduction of semantic inference in event processing (by realizing the so-called *Semantic Complex Event Processing* (SCEP) [13]), in order to obtain a knowledge-based detection of complex event patterns that goes beyond what is possible with current solutions. To this aim, we designed an agent that dynamically builds up a Resource Description Framework (RDF) [14] ontology, based on the incoming events, and applies queries expressed in the SPARQL query language [15] for semantic