Analysis of Big Data Architectures and Pipelines: Challenges and Opportunities

Álvaro Valencia Parra

Supervisors: Ángel Jesús Varela Vaca
María Teresa Gómez López

Departamento de Lenguajes y Sistemas Informáticos
Escuela Técnica Superior de Ingeniería Informática
Universidad de Sevilla

Máster Universitario en Ingeniería Informática

February 2019

A mis padres por su cariño, sus consejos, y el apoyo incondicional que me brindan en cada paso que doy en mi vida.

Acknowledgements

I would like to thank my supervisors, Ángel Jesús Varela Vaca and María Teresa Gómez Lopez for their continuous support and good advice during the development of this work.

This work has been partially funded by the Ministry of Science and Technology of Spain (TIN2015-63502-C3-2-R) and the European Regional Development Fund (ERDF/FEDER). The author would like to thank the Cátedra de Telefónica of the Universidad de Sevilla for its support.

Abstract

The continuous technological advances are promoting changes in multiple aspects of society. One of the consequences of these advances is the increase in the amount of data that is daily generated. In this scenario, Big Data has emerged as one of the most disruptive paradigms in recent years, becoming a matter of great interest for multiple types of organizations. This interest is due to the fact that Big Data is enabling organizations to extract value from the data they own. At the same time, Big Data is promoting more technical changes that are increasing the potential value that can be extracted from data. This value enables companies to increase and optimize their productive capacity, contributing to increase their competitive advantages, and to ease the decision making process.

As a result, Big Data has become one of the most studied fields, both in literature and in Industry. Consequently, it is constantly evolving, and presents significant challenges and opportunities that could increase the quality of the process of value extraction from data. However, since the Big Data paradigm is continually evolving, a detailed and concise study about all aspects related to it is required.

In this work, a research about the state-of-the-art of the Big Data paradigm is carried out. The concepts related to it, the activities and techniques on the value extraction process, and the data processing architectures are studied. Next, the main limitations, challenges, opportunities, and possible research lines related to the Big Data paradigm are identified. Finally, a solution to one of the research challenges that arise in this study is proposed: a framework to deal with the preparation of data with complex structures.

Resumen

Los continuos avances tecnológicos están promoviendo cambios en múltiples aspectos de la sociedad. Una de las consecuencias de estos avances y cambios sociales es el aumento de la cantidad de datos que se generan día tras día. En este escenario, Big Data ha emergido como uno de los paradigmas más disruptivos de los últimos tiempos, siendo de gran interés para múltiples tipos de organizaciones. Este interés se debe a que Big Data está permitiendo a las organizaciones a extraer valor de los datos que tienen a su disposición. Al mismo tiempo, Big Data está promoviendo más cambios tecnológicos que están aumentando el potencial valor que se puede extraer de los datos. Este valor permite a las empresas aumentar y optimizar su capacidad productiva, contribuyendo a la mejora de sus ventajas competitivas, y facilitando la toma de decisiones.

Como consecuencia, Big Data se ha convertido en uno de los campos más estudiados, tanto en la literatura como en la Industria. Se trata de un campo que está en continua evolución y que presenta unos retos y oportunidades muy sustanciales que podrían aumentar la calidad del proceso de extracción de valor de los datos. Sin embargo, al ser un campo en continua evolución, se requiere un estudio detallado y conciso de todos los aspectos relacionados con este.

Este trabajo realiza un estudio sobre el estado del arte y los conceptos relacionados con el paradigma Big Data, las actividades y técnicas relacionadas con el proceso de extracción de valor de los datos, y las arquitecturas de procesamiento de los mismos. Este estudio se estructura en tres partes. En la primera, se contextualizan los conceptos y actividades relacionadas con el paradigma Big Data, proponiendo una visión global de este. En segundo lugar, se identifican las principales limitaciones, retos, oportunidades, y posibles líneas de investigación relacionadas con el paradigma Big Data. Por último, se propone una solución a uno de los retos de investigación que se plantean en este estudio: la preparación de datos con estructuras complejas.

Table of contents

List of figures

List of tables

Chapter 1

Introduction

1.1 Context and Motivation

The industrial revolutions that took place throughout history and the following technological improvements had a high influence on people's habits: the way they work, use their free time and do their daily tasks evolved. Industry, technology, and society advanced as these changes were introduced. As a result, the information generated by humanity tends to increase exponentially [2]. Two consequences arise in this regard. On the one hand, as the amount of data increases, the techniques employed to store and process it become more sophisticated. On the other hand, as technology improves, the potential value of the data increases as well. In this respect, Peter Sondergaard, vice president of the consulting firm Gartner between 1988 and 2018, claimed that data is the oil of the 21st century [3].

Since the Sixties of the last century, systems to store digital data have been developed. Database management systems were employed by companies to store operational data and perform analytic operations based on descriptive statistics, known as Business Intelligence. Although traditional database systems were useful and are still used, these have limitations in some use cases. The emergence of new technologies, the widespread use of the internet, and the growing impact of the digital world on people, led to the emergence of a type of data that fulfilled three fundamental characteristics: they were generated in bulk (volume), at high speed (velocity), and had different semantics and structures (variety). The Big Data paradigm managed to overcome the limitations that traditional database systems had when dealing with this kind of data. This paradigm is meant to capture, store, manage and analyze data which fulfill the characteristics mentioned above (volume,

velocity and variety). Big Data Pipelines enable to the extraction of value of such data by defining a set of activities.

In recent years, Industry is facing a new revolution, based on digital transformation [4] (i.e.,the digitalization and automation of business processes). Big Data is the cornerstone of this digital transformation since it enables companies to analyze large amounts of data generated not only by internal sources, but also by external ones. Other pillars on which this digital transformation is based are: (i) the Internet of Things (hereinafter, IoT), which allows companies to capture data from most of their business processes; and (ii) the use of robots and autonomous systems to perform tasks without human interaction.

For all the reasons mentioned above, Big Data is a field of great interest, widely studied, and on which great innovations are made year after year. Consequently, there are still many concepts, challenges, and opportunities to explore. These could significantly improve the way in which companies benefit from this paradigm. For instance, there are numerous limitations in the real-time data analysis, the processing of data with complex structures, or the analysis of the quality of data.

1.2 Main Contributions

The objective of this work is to carry out a study about the state-of-the-art of Big Data, Big Data Pipelines, and the components of a typical Big Data Pipeline. This work has been carried out by researching studies and proposals that other authors have recently made in the literature. As a result, several research findings have been identified. In addition, a proposal is made based on the processing of data with complex structures in Big Data environments. In summary, this work intends to provide the following value in the field of Big Data:

- The contextualization of concepts and activities related to Big Data and Big Data Pipelines given in the literature.
- The definition of a global schema which covers the activities related to a Big Data Pipeline.
- The identification of the main limitations, challenges, opportunities, and possible lines of research regarding the Big Data field.
- A solution for the transformation of complex data structures in Big Data environments.

1.3 Roadmap

The remaining of this work is structured as follows: Chapter 2 includes a study on the most important concepts about Big Data that can be found in the literature. Chapter 3 devises the research findings about the most challenging concepts in the Big Data context. Chapter 4 depicts the proposed approach about the transformation of complex data structures in Big Data environments. Finally, Chapter 5 concludes this study and draws future work.

Chapter 2

Foundations

2.1 Introduction

Big Data is an emerging area in the Information Technology field. As mentioned previously, it is one of the most disruptive and challenging technologies which are transforming the way in which businesses devise their strategies, objectives, relationships with the competence, providers, customers and so on. Therefore, it is crucial to understand the concept of Big Data, the underlying architectures, technologies, and even the tasks that might encompass a Big Data solution.

This chapter introduces the concept of Big Data in Section 2.2, being in Section 2.3 where the most relevant concepts, tasks and activities of a Big Data Pipeline are defined as a framework which binds the activities of the pipeline together. Finally, Section 2.4 gives an overview of the most relevant Big Data architectures according to the literature and Industry.

2.2 The Concept of Big Data

There is no a consensus about what is the definition of a Big Data. In 2011, the McKinsey Global Institute defined Big Data as "datasets whose size is beyond the ability of typical database software tools to capture, store, manage and analyze" [5]. They do not define a minimum size to consider whether a dataset is Big Data or not. They say that this concept is relative to the available software tools, so it depends on a particular moment in history and on "what sizes of datasets are common in a particular industry". This definition implicitly defines the steps in a Big Data

process: *capture*, *store*, *manage* and *analyze*. This process will be discussed in this chapter as part of the *Big Data Pipeline* section.

Many authors agree with the definition provided by the McKinsey Global Institute. For instance, E. Curry et al. [6] say that "Big Data brings together a set of data management challenges for working with data under new scales of size and complexity". A. Oussius et al. [7] include the data heterogeneity in this definition, pointing out the "complex nature" of Big Data, which "require powerful technologies and advanced algorithms". This complex nature of Big Data was already defined in 2001 when D. Laney forecast the massive data management problem in the next years. It was characterized by *the three dimensions of Big Data* [8] [9] (a.k.a the 3 Vs of Big Data):

- **Volume**. It refers to the amount of data which an organization is able to generate and process. As discussed above, the volume of data to be considered Big Data varies depending on the software tools capabilities and the type of industry. This volume of data is not only generated by operational databases but also by sensors, social networks, and other external data sources [7].

- **Velocity**. It refers to "the speed at which data are generated an processed" [9]. This speed is motivated by the amount of data sources available for an organization and the frequency with which they generate data. The same authors mentioned that the speed at which the data is generated is a challenge. Traditionally, batch processing has been used to deal with Big Data, but nowadays streaming processing is being used to face the velocity challenge.

- **Variety**. It refers to the type of data which an organization generates. This variety is not only related to the data structure, but also to the semantic of this data. It means that Big Data includes structured or semi-structured data, or unstructured data, such as plain text, logs, images or videos [7] [9]. Big Data also includes data from different domains, arising the challenge of dealing with data with different semantics.

Some authors include additional dimensions regarding the Big Data complex nature. Below, four of them are shown.

- **Veracity**. This dimension was proposed by IBM™[9], representing the "unreliability and uncertainty latent in data sources", motivated by the "incompleteness, inaccuracy, latency, inconsistency, subjectivity and deception" in data.

Other authors agree with this definition. For example, D.P. Acharjya [10] includes the availability and accountability in the veracity definition; M. Obitko [11] includes the ambiguity, pointing at the importance of the veracity in Big Data due to the velocity and variety dimensions (the more data is generated and the more varied it is, the more issues in veracity may occur).

- **Variability and Complexity**. These dimensions were proposed by SAS™[9]. The first points at the "variation in data flow rates". It means that the frequency with which data is generated and processed is not uniform, varying depending on chaotic factors. The second one refers to the complexity derived from "the need to connect, match, cleanse and transform data received from different sources" [12]. Both dimensions are strongly related to the velocity and variety dimensions, and are becoming an important challenge in the Big Data context. S. Nadal et al. [13] extends the definition of variability, including the variation in the data itself (e.g., the schema and semantics) and changes in the data sources.

- **Value**. This dimension was proposed by Oracle™[9]. It refers to the fact that normally raw Big Data tends to be low-valued. However, in a holistic view, the value tends to be higher [12].

- **Decay**. It refers to the loss of value over time [9]. It becomes quite important in contexts of high velocity, when data is required to be analyzed in near real-time.

Big Data enables organizations to extract value from it. Organizations that manage to extract knowledge from their data will have many competitive advantages [6] by obtaining an in-depth understanding of the business [14], supporting the decision making and innovation processes. Consequently, they will be able to deliverer better products [15]. The set of activities which enables organizations to extract value from Big Data is the Big Data Pipeline. Nevertheless, these activities are challenged by the Big Data dimensions, and they must be faced up in a Big Data Pipeline. In the next section, the process of extracting value from data is discussed, and a holistic view of Big Data Pipelines is provided as well as a detailed description of every activity. The architectures which enable to process Big Data are discussed in Section 2.4 .

2.3 Big Data Pipeline

Big Data Pipeline includes the activities that enable companies to extract value from their data. Several proposals and definitions about Big Data Pipelines and their activities can be found in the literature. In this section, we propose a pipeline and a definition for each activity of the Big Data Pipeline by taking into account various studies that have been carried out in the literature.

P. Ceravolo et al. [16] define the concept of pipeline as "the coordination of different tasks, integrating different technologies, to achieve a specific solution". In the Big Data context, the objective of those tasks is "to drive Big Data computations". These authors propose six *stages*: *data acquisition and recording*, *data extraction and annotation*, *data preparation and cleaning*, *data integration and aggregation*, *data processing and querying*, and *data interpretation and reporting*. Different tasks in relation to the data quality are carried out between one activity and the others.

C. Ardagna et al. [17] propose a pipeline to support a model-driven methodology oriented to Big Data services. Five high-level areas are proposed: *data preparation*, *data representation*, *data analytics*, *data processing*, and *data visualization and reporting*.

P. Paakkonen2015 et al. [18] come up with a pipeline to represent the typical data flows and the activities in Big Data processes. The activities of this pipeline are: *data extraction*, *data loading and pre-processing*, *data processing*, *data analysis*, *data loading and transformation*, and *interfacing and visualization*.

A concept strongly related with the idea of Big Data Pipeline is the *Big Data Value Chain*, introduced by E. Curry as "the information flow within a Big Data system as a series of steps needed to generate value and useful insights from data" [6]. Five activities compose this value chain: *data acquisition*, *data analysis*, *data curation*, *data storage*, and *data usage*. In this work, we will consider the Big Data Pipeline as an enabler for this value chain.

These definitions concur that a Big Data Pipeline is a process composed of a set of activities whose objective is to extract value from data. According to the pipelines proposed in the literature and the definition of the activities that compose them, a global framework for Big Data Pipelines is proposed. This is represented in Figure 2.1 and it is composed of six major activities that are described below.

- **Data Acquisition**. It is intended to collect the information from data sources, and to ingest data in order to transport it to the next activity in the pipeline. A prior quality filter can be applied before the ingestion. P. Ceravolo et al. [16] include this objective in the pipeline that they propose as the *data acquisition*

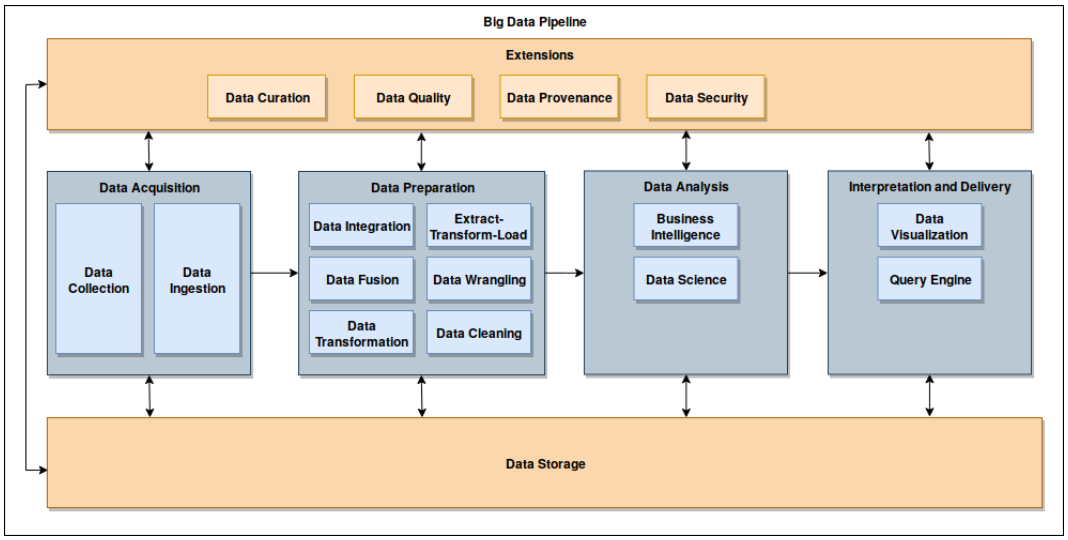


Fig. 2.1 Big Data Pipeline schema.

and recording stage. P. Pääkkönen et al. [18] represent the data acquisition as the act of collecting the data from the data source in the *data extraction* activity. The ingestion is implicitly performed in both *data extraction* and *data loading and pre-processing*. E. Curry [6] also includes this activity in the Big Data value chain.

- **Data Preparation**. The objective of this activity is to prepare data for its processing in the next activities by formatting, cleaning or fixing it. Six tasks implement techniques which enable data preparation: data integration, data fusion, data transformation, Extract-Transform-Load, data wrangling and data cleaning. P. Ceravolo et al. [16] propose three stages which conform it: *Data Extraction and Annotation* for the data formatting and restructuring, *data preparation and cleaning* in order to clean, enrich and improve the privacy of data, and *data integration and aggregation* for integrating and standardizing data from several sources. C. Ardagna et al. [17] consider two areas which have similar objectives to those previously exposed: *data preparation* and *data representation*. P. Pääkkönen et al. [18] propose the *data processing* activity, which have similar objectives to the data preparation as has been defined. E. Curry [6] includes these tasks into the *data curation* and *data analysis* activities.

- **Data Analysis**. The data analysis intends to extract value from data by mining it. Both Business Intelligence and Data Science can be applied in order to reach it. P. Ceravolo et al. propose the *data processing and querying* stage, whose

objective is "querying and mining Big Data". C. Ardagna et al. [17] include these objectives in the *data analytic* area. Both P. Pääkkönen et al. [18] and E.Curry include a *data analysis* activity in their proposals.

- **Interpretation and Delivery**. It is intended to be the final activity in a Big Data Pipeline. It aims to report the value extracted from data through the pipeline to benefit the business activities which require it. P. Ceravolo et al. [16] include the *data interpretation and reporting* activity which conforms these goals. C. Ardagna et al. [17] propose the *data visualization and reporting* area as the one responsible for the representation of the results. P. Pääkkönen et al. [18] consider two activities: *data loading and transformation*, which is intended to transform and transfer the results of the analysis, and *interfacing and visualization*, which is responsible for visualizing and reporting the results. E. Curry [6] mentions the *Data Usage* activity with the objective of integrating the results of the data analysis within the business activity.

- **Extensions**. It comprises a set of activities which can be carried out in parallel with the other activities. The extensions included here must guarantee the quality, security and legal requirements over the whole process. Three sub-activities have been identified: data quality, data provenance, data curation, and data security. P. Ceravolo et al. [16], C. Ardagna et al. [17] and E.Curry [6] implicitly include data quality, provenance and security tasks in their proposals.

- **Data Storage**. This is also a transversal activity. Its objective is to persist and provide access to the data when required. P. Ceravolo et al. [16], C. Ardagna et al. [17], P. Pääkkönen et al. [18], and E.Curry [6] also perform storage tasks in their proposals.

There are some activities in the previously cited pipeline definitions which are not covered by the Big Data Pipeline which have been proposed above. One of them is the *data processing*, proposed by C. Ardagna et al. [17]. In accordance with these authors, this area is intended to manage both the data flow and the parallelization. Nevertheless, in this work, these aspects of Big Data will be considered as architectural issues that are discussed in Section 2.4.

The next subsections analyze the impact of the Big Data dimensions in each activity of the pipeline.

2.3.1 Data Acquisition

The data acquisition concept has been defined by several authors. K. Lyko et al. [19] define it as the process of "gathering data from distributed information sources with the aim of storing them in scalable, Big Data-capable storage". M. Shah [20] uses this concept to define the acquisition of data from edge devices in IoT contexts with the objective of transferring it through a network. S. Poornima et al. [21] hold that data acquisition is a process which comprises the following phases: (i) data collection from "real world objects"; (ii) data transmission into a storage system; and (ii) data pre-processing. P. Ceravolo et al. [16] include the need of interpreting the source as well as filtering data depending on its relevance.

There is not a specific consensus about the objectives of data acquisition. While some authors concentrate on the collection of data, others include the transmission to another stage or even the pre-processing of it. For this reason, in this work two phases are considered in regard to the data acquisition process:

- **Data collection**. It comprises the collection of data from data sources such as databases, edge devices, etc.
- **Data ingestion**. It comprises the delivery of the collected data so that it can be pre-processed or processed in a data pipeline.

Data acquisition entails some challenges itself. Regarding the Big Data dimensions, the process of data collection and ingestion must deal with high volumes of data at high velocities and wide varieties of formats and schemata [19]. It becomes especially critic in IoT contexts [20], where edge devices continuously collect information with heterogeneous format and semantic.

In order to enable data acquisition, a framework to collect data from distributed data sources is required [19]. Data sources may use different protocols in order to facilitate the information retrieval from them. The framework must be able to use these protocols so that they can communicate and interpret the information given by the data sources and transfer it to the next step in the pipeline.

Data acquisition also challenges the veracity, variability and value dimensions. There are still research issues regarding the relationship between data acquisition and these Big Data dimensions that are considered and discussed in Chapter 3.

2.3.2 Data Preparation

There are several definitions of data preparation in the literature. C. Ardagana et al. [17] say that data preparation includes the activities whose aim is to prepare data for analysis. While G. Mansingh et al. [22] state that data preparation consists of "identifying quality data and formatting it appropriately". They identified three tasks: cleaning the data, constructing the data, and transforming the data format. The task related to the construction of the data is intended to calculate derived properties, discretize variables and carry out data integration when required. In Industry, data preparation is defined as the process of transforming raw data into a clean dataset [23]. Data cleaning, data integration and data transformation are part of this process. The data preparation requires a high dependence on both data scientists and experts in the problem domain [22]. This dependence makes this activity one of the most error-prone and time-consuming tasks in the Big Data Pipeline.

By taking into consideration these definitions, we propose the following definition of data preparation: the process which aims to transform raw data into a clean dataset in order to facilitate its consumption, considering a clean dataset the one which fits a specific schema, format, and a set of quality, security and privacy rules.

There is a set of techniques which can be found in the literature and Industry based on (i) the modification of the data structure and/or the value of data or (ii) data filtering. These techniques have been collected and depicted in Figure 2.2.

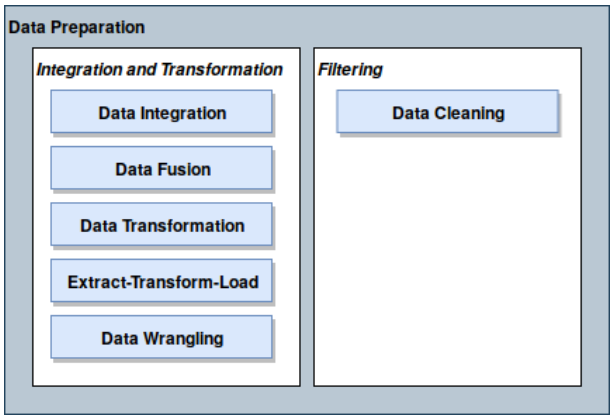


Fig. 2.2 Data preparation tasks.

Big Data dimensions have an impact in the use of data preparation in the Big Data context. In this stage, data may come from thousands of different sources with heterogeneous schemata at non-uniform high speed rates, generating large

amounts of data. Advanced algorithms and computation capabilities are required to overcome the challenges derived from the volume and velocity dimensions. Variety dimension is an important stumbling block, especially in the context of data integration, where many data schemata must be homogenized. Other dimensions (e.g., veracity, variability and complexity) have an important impact in the data preparation. The veracity is crucial to succeed in the other steps of the Big Data Pipeline. Data preparation must make an effort to improve the quality of data during the process of integration and transformation by formatting and filtering data. Variations in data flow rate and the need to integrate them imply that the variability and complexity dimension have influences in the preparation process.

Data Integration and Data Fusion

X. Dong et al. [24] define data integration as the process which has the goal of "providing unified access to data residing in multiple, autonomous data sources". B. Arputhamary et al. [25] define data integration as a process which consists of the data transfer from a source schema to a target one. In Industry, data integration [23] is considered the process which enables to combine data from different sources to conform *consistent data*.

According to X. Dong et al. [24], three steps conform the data integration process: (i) schema alignment, (ii) record linkage and (iii) data fusion.

- **Schema alignment**. It is intended to solve the problem that arises when the same reality is represented with different schemata. In order to solve it, it must identify attributes with the same semantic in every schema.
- **Record linkage**. Once aligned the semantics of attributes in source schemata, it is necessary to unify the format of the attributes which have the same semantic. X.Dong et al. [24] point at typo errors and multiple naming conventions as examples of differences between value format. Differences between date formats, and numeric representations are also examples of it.
- **Data fusion**. It is intended to solve conflicting values which remain after the record linkage step due to "inconsistencies in the interpretation of the semantics, outdated information, incorrect calculations [...]". It does so by selecting the true value between a set of inconsistent values for a given attribute.

Regarding the data fusion, it has two different meanings in the literature. Y. Zheng [26] defines both of them. The first one considers data fusion in the context

of data integration in the same way as X. Dong et al.: "data fusion is a process of integration of multiple data representing the same real-world object into a consistent, accurate, and useful representation". The second definition defines data fusion as a process to combine data from totally different domains so that data analysis techniques can combine knowledge from several domains. While this process is similar to data integration (both of them combine data from multiple sources), there are some differences between them. The most important is that the data integration is meant to process data in a batch-oriented paradigm, while data fusion is stream-oriented, being specialized in time-series data. Data fusion also uses data reduction techniques in order to simplify and increase the relevance of data [27] [28].

Both data integration and fusion are affected by the three main Big Data dimensions. Algorithms and techniques must deal with the volume and variety of data. Data preparation in general, and data integration and fusion in particular, are the activities in which the variety dimension has the greatest impact. As mentioned in this section, data integration techniques are specially designed to unify different schemata and data formats. Therefore, a high variety complicates the integration process. The veracity dimension also has an important impact in the integration and fusion process due to data inconsistencies, different formats, etc. Regarding the variability and complexity dimension, data fusion is the most challenged due to its stream-oriented condition.

Data Transformation

P. Cong et al. [29] define data transformation as the conversion, reformatting, computation and rebuilding of data. In industry, data transformation is considered the process which "convert raw data into a specified format according to the need of a model" [23]. This reference mentions three tasks in the data transformation: normalization, aggregation and generalization.

According to these definitions, in this work data transformation is defined as a process which includes the schema conversion, the transformation of existing attributes and the creation of new ones. This process is applied to unstructured, semi-structured and structured data schemata. The transformation process can be carried out as a part of other data preparation tasks, such as data integration, Extract-Transform-Load or data wrangling.

Data transformation is mostly affected by the variety dimension since the existence of complex data structures with nested array and schemata makes this process non-trivial.

Extract-Transform-Load

The Extract-Load-Transform (ETL) [30] paradigm has been traditionally used to deal with data and schema transformation in the data-warehouse context, enabling analysts to perform data extraction from heterogeneous data sources, data cleansing, data formatting, data integration, and data insertion. Although this is an old technique widely used before the Big Data paradigm emerged, this technique is still useful in the Big Data context to prepare data for its consumption by easing tasks related to the integration of data from different sources by unifying the data schema, the modification and formatting of data to uniform its format, and the unification of data in a single stream. One of the most significant use cases in the Big Data context is the preparation of data to perform Business Intelligence analytics. Nevertheless, ETL presents some limitations [31] which can be solved by combining this technique with other data preparation techniques. The lack of support for unstructured or semi-structured data schemata, and the lack of techniques to improve data quality are examples of these limitations. Another one is that this technique requires high IT technical acknowledgement and programming skills.

Data Wrangling

In recent years, a new paradigm for data preparation called *data preparation self-service* or *data wrangling* has emerged [32]. The idea behind it is that non-expert users can carry out data preparation tasks, being able to deal with unstructured, semi-structured and structured data coming from multiple data sources. This paradigm solves some problem derived from traditional data preparation techniques, contributing to the reduction of the implication of IT-experts. In this way, the gap between domain experts requirements and technology is smaller [33].

Data wrangling has become the cornerstone of this new paradigm. It enables non-expert users to transform raw data into the format required by the data consumers [23]. For instance, The By-example paradigm [34] [35] is being used to facilitate the data transformation task by simply providing examples on how is the input format and how would be the output.

In accordance with J. Hellerstein et al. [33], data wrangling consists of the following tasks:

- **Discovery and assessment**. This task is expected to find out basic information about the nature of the data (e.g. its structure, its format, etc.).

- **Structuring**. It is intended to perform the required data transformations which imply a change in its structure.
- **Cleaning**. It consists of the filtering of low-quality data.
- **Enriching and blending**. It aims to improve the consistency, accuracy and completeness of the data.
- **Optimizing and publishing**. It consists of the optimization of the data structure in order to make it more suitable for its consumption and the transmission to the next step in the pipeline.

Data wrangling is affected by the aforementioned Big Data dimensions. In this case, the variety is critic, because it strongly conditions the structuring task. Nevertheless, this process might be benefited from the volume of Big Data due to the need of collecting information about the data and its context to facilitate these tasks, leading to an improvement in the quality of the results [35].

Data Cleaning

P. Cong et al. [29] said that the objective of the data cleaning process is to "clear wrong data values and redundant records [...] according to a set of cleaning rules [...]. The ultimate goal of data cleaning is to improve data quality". Data cleaning is hence an important component of the quality assurance process. Although most authors place this task in the data preparation context [36] [24] [29], others place it in further activities of the pipeline. For example, E. Curry [6] incorporates data cleaning in the data acquisition activity.

2.3.3 Data Analysis

Data analysis is the key activity in the Big Data Pipeline. It is meant to extract value from data. Depending on the objective of the analysis and on the type of value to extract from the data, analytic techniques can be descriptive, predictive, or prescriptive [21].

- **Descriptive Analysis**. Data is used to identify a particular phenomena and to analyze its reasons.
- **Predictive Analysis**. Data is used is to predict future events by using historical data.

- **Prescriptive Analysis**. It tries to make the best decision to take advantage or mitigate either a known or a predicted fact calculated by using descriptive or predictive analysis, respectively.

Traditionally, and even before the Big Data paradigm burst into the scene, organizations had been performing Business Intelligence in order to extract value from their data. When Big Data came into scene, it enabled a new disciplined whose mission was to extract further value from data: the Data Science [37]. While both are intended to extract value from data, there are some differences between them.

Business Intelligence

The term Business Intelligence became popular in companies in the 1990s [38]. Traditionally, it has been based on the descriptive analysis on structured operational data sources to support decision-making by answering questions about what have been happening in the organization [39].

Both descriptive statistics methods and query engines are enablers of Business Intelligence, allowing businesses to find out data which fit a particular pattern or a set of them. In order to perform queries, a structured data model is needed. The data model must be able to answer the questions related to a business report [40]. In general, the queries are typically written in SQL language.

In short, Business Intelligence extracts value from data by answering questions about what happened in the past. These are answered in a business report that is built by performing queries and applying descriptive statistical methods on a structured data model.

Business Intelligence is mostly impacted by the volume and velocity Big Data dimensions. Since queries and non-computational-complex calculations are typically required, the amount of data to process and the speed in which it is generated and demanded are key factors in the efficiency of a data analysis. On the other hand, the veracity dimension also impacts the Business Intelligence process. As B. Schmarzo [41] states, the accuracy of the data is a critical aspect in order to obtain high quality reports.

Data Science

Data Science [37] is a discipline that became popular when Big Data came into scene. Therefore, Big Data was in fact an enabler for data science. According to V. Dhar et al. [42] Data Science is "the study of the generalizable extraction of knowledge from

data". The kind of knowledge that is extracted from data consists of patterns which fit the observations (i.e., the data), unlike Business Intelligence, which is meant to find out the observations (i.e., the data) which fit a pattern or a set of them. Here is where the predictive and prescriptive analysis come into scene.

From the point of view of a company, data science tries to answer two kinds of questions, depending on the type of analysis to apply [43]: "What is likely to happen" (it is answered by applying predictive analysis) and "What should we do" (this question is answered by using prescriptive analysis).

Another key difference from Business Intelligence, is that one of the Data Science fundamentals is the need to deal with unstructured data unlike Business Intelligence, which as mentioned above, it was thought to deal with structured data.

Many techniques to carry out data science have been developed. Machine learning algorithms are one of the most employed techniques to perform predictive and prescriptive analytics. It is meant to (i) discover knowledge and (ii) make decisions automatically [7]. These algorithms must be trained by using a dataset. The training process infers patterns in data and creates a model. The model is employed to select (predict) the pattern which better fits new instances of data. Machine learning algorithms can be classified in different ways depending on (i) what type of value must be predicted; and (ii) how is the dataset used to train the algorithms [44].

On the one hand, depending on the type of value to predict, there are two major types of algorithms:

- **Regression algorithms**. These type of algorithms are meant to predict continuous values.
- **Classification algorithms**. These type of algorithms are meant to predict discrete values.

On the other hand, depending on how is the dataset which is employed to train the algorithm, there are two major types of them:

- **Supervised learning**. Each tuple of the dataset has the observation and the value that should be returned by the algorithm.
- **Unsupervised learning**. Each tuple of the dataset only has the observation. The algorithm must find out common patterns among the data and assign them a value.

In recent years, a disruptive technique to perform predictive and prescriptive analytics has emerged: Deep learning. These algorithms are able to outperform machine learning algorithms in many cases (e.g., analysis of images, texts, and similar) [45]. However, deep learning entails more computational complexity, and therefore, it requires high computational resources in order to work efficiently.

Prescriptive analysis can also be performed by using Constraint Optimization Problems. This technique can be used to support the decision-making process by analyzing either organizational datasets or the results of a predictive analysis. The use of this technique in Big Data environment is still a research issue. L. Parody et al. [46] proposed an approach in that direction.

In regard to the Big Data dimensions, Data Science is more conditioned than Business Intelligence regarding the volume and velocity dimensions due to the computational complexity of the algorithms. The veracity dimension also as an impact in data science. According to B. Schmarzo [41], the quality of data conditions the results of the analysis.

2.3.4 Interpretation and Delivery

Interpretation and delivery intends to be the final step in a Big Data Pipeline. Its objective is to deliver the value extracted from the pipeline by interpreting and exposing the results to either a final user or a service.

Other authors have included similar steps in the Big Data Pipeline. For example, P. Ceravolo et al. [16] included the *data interpretation and reporting* step in their proposal. The objective of it step is to perform an exhaustive interpretation of the results in order to "verify the assumptions that allow drawing safe conclusions". In the context of the Big Data value chain, T. Becker [47] defines an activity (i.e., *data usage*) of which objective is to give support to business activities which needs to access data and the results of the analysis, among others. The ultimate goal is to facilitate the decision-making activity.

As the final step in a Big Data Pipeline, it must adapt the results of the data analysis in order to enable its usage. The way the results are interpreted and delivered strongly depends on the consumer and the ultimate objective of the whole Big Data Pipeline. For instance, if the data analysis activity returned values about predictions, the results must be properly formatted and presented to a business manager to facilitate the decision-making. In this case, a report or a query engine may be suitable to achieve this objective. However, if the consumer is a service such

as a cyber-physical system, the predictions must be interpreted so that this service can understand and act in accordance with the prediction.

To summarize, the possible consumers of the interpretation and delivery activity are listed bellow. The tasks that might be necessary in order to fulfill the consumer requirements are specified.

- **Final User**. A human person such as a business manager or employee may be the consumer of the interpretation and delivery step. Results may be interpreted and delivered to this final user by means of a data visualization tool and a query engine. On the one hand, the visualization tool can give support to the interpretation of the results by giving the possibility of creating custom reports. On the other hand, a query engine can help to the final user if the pipeline is generating a data flow by supporting the search for results that fit some conditions.

- **Service**. The consumer of the interpretation and delivery step may be a non-human entity. The Big Data Pipeline is expected to give support to a service such as other Big Data Pipelines, monitoring systems, information systems or cyber-physical systems. These services can consume the pipeline results by means of an Application Programming Interface (hereinafter, API) of which objective is to properly interpret the analysis result and to enable its consumption.

This step is especially challenged by the velocity and volume. The velocity can be challenging when the pipeline returns a data flow. The interpretation and delivery process must be capable of satisfying both the frequency with which results are generated and the demands of consumers. Regarding the volume, visualization tools, query engines and APIs must deal with a scenario in which large amounts of result data are generated. Other dimensions such as the veracity, variability and complexity can be a challenge. On the one hand, the results of the data analysis could be tested in order to verify their quality and relevance. On the other hand, if the pipeline returns a data flow, the variation in the speed rate at which results are delivered can challenge the interpretation and delivery process.

2.3.5 Data Storage

Data storage is one of the fundamental pillars of the vast majority of Big Data Pipelines. In this context, the data storage provides both reading and writing capabilities in any other pipeline activity.

The volume, velocity and variety dimensions have a direct impact on the data storage activity. Storage technologies must be capable of storing large volumes of data, satisfying the demand for reading and writing, and supporting data with heterogeneous structures. To address these challenges, different types of storage systems suitable for Big Data environments have been developed. These systems typically use distributed data storage, implement complex search engines and support real-time queries [48]. An example of a distributed data storage is the *HDFS* paradigm [49]. On the other hand, to deal with heterogeneous data, different paradigms specialized in different types of data have been developed. An example is the NoSQL paradigm, which supports semi-structured data.

2.3.6 Data Extensions

The Data Extensions are key activities in a Big Data Pipeline. These are carried out during the whole life-cycle of the pipeline with the aim of ensuring quality, security and privacy requirements. While most of them have direct or indirect impact on the entire Big Data Pipeline, others are focused on some specific steps of it. These activities are described bellow.

Data Quality

Data quality [50] is a condition of data which is assessed by using a set of variables called *data quality dimensions*. The data quality task is meant to monitor and measure such condition. I. Taleb et al. [36] indicated that data quality is mostly employed in data preparation. They also said that the quality depends on the problem domain, the methods and the measures employed to assess it. The ISO standard ISO/IEC 25012 [51] states that data quality is the degree to which data satisfies the requirements of the user. These can be classified in a set of characteristics or dimensions. The data quality model devised in this standard classifies them in two types: (i) inherent data quality, which refers to those dimensions that assess the quality depending on the "intrinsic potential" of data needed to satisfy a set of quality requirements (e.g., accuracy, completeness, consistency, credibility, and currentness); and (ii) System-Dependent data quality, which considers those data quality dimensions which are

dependent on the technological context in which data is processed (e.g., availability, portability, and recoverability). In this study, inherent data quality dimensions are considered in accordance with the definitions given in the standard ISO/IEC 25012 [51]:

- **Accuracy**. It is the degree in which data attributes precisely represent the true value of its attributes.
- **Completeness**. It is the degree in which there are enough values for all the attributes so that it can faithfully represent the event.
- **Consistency**. It is the degree in which the data is not contradictory neither with itself nor with other data.
- **Credibility**. It is the degree in which data is trustworthy and credible for users.
- **Currentness**. It is the degree in which data is consistent with the temporal context of its creation.

Data quality is a major problem in the Big Data context. As A. Freitas et al. [52] state, the quality of the results in a data analysis process depends on the quality of the data employed in that process, which has a significant impact in business operations such as decision-making. In this regard, A. Siddiqa et al. [53] also claimed that poor quality data is a problem for organizations, and techniques to detect and clean this type of data are needed. Nowadays, the quality of captured data tends to decrease as unstructured data are generated at high scale [54].

Data quality is intended to deal with the veracity Big Data dimension by maximizing the reliability, certainty and consistency of data. Nevertheless, the data quality assurance process is conditioned by other Big Data dimensions such as the volume, velocity and variety [36]. Consequently, data quality algorithms must be capable of processing large amounts of heterogeneous unstructured, semi-structured or structured data at high speeds.

Data Provenance

Data provenance is the process of finding out the origin, the context in which data was created, and all the transformation process that generated that data [55] [16]. C. Ardagna et al. [56] remarked the importance of the data provenance as a key aspect in the Big Data since it is meant to increase the reliability of data analysis.

Thanks to data provenance, both final users and other Big Data services or activities can benefit from this information. For instance, data quality processes can be significantly improved by means of the data provenance. As E. Freitas et al. [52] stated, "some data quality attributes can be evident by the data itself, while others depend on an understanding of the broader context behind the data". The provenance, processes, and actors which take part in the data generation are examples of the context of data which can be used to assess the credibility of both the data and the whole process. These authors define data provenance in the context of data curation as a cornerstone for providing the required context to select reliable data. They also stress that the decisions taken in the curation process must be captured an registered by the data provenance.

Another example is the security and privacy. In this context, data provenance can help to improve dimensions such as the integrity, the availability, the confidentiality, and especially the privacy.

As mentioned above, data provenance is a cornerstone to improve the quality of the services provided by Big Data, and hence, it takes part in the whole Big Data Pipeline. The following list describes the impact of data provenance on the remaining steps and activities on a Big Data Pipeline.

- **Data acquisition**. Data provenance must record metadata related to the collection and ingestion of the data (e.g., the data source, the moment in which the data was generated, the author, the channels used to transfer the data, the filters and transformations applied to data during its ingestion, and others).

- **Data preparation**. As mentioned above, keeping track of the transformation applied to data is a major issue in the data provenance process. At the same time, it must provide information to the data quality process to facilitate data curation, cleaning, wrangling, transformation, integration and fusion tasks.

- **Data analysis**. The data provenance must record all the operations carried out to the data during its analysis [57]. Nowadays, advances in data science are pointing out [58] the need to understand machine learning and deep learning models and find out why they make certain decisions. Data provenance is key in this new paradigm.

- **Interpretation and delivery**. If required, the data provenance must be interpreted and delivered so that it can be consumed.

- **Data Storage**. It must be able not only to store the metadata, but also to maintain the relationship between the data itself and its corresponding metadata.

In the Big Data context, data provenance is strongly impacted by the volume, velocity and variety dimensions. Dealing with large amount of heterogeneous data is still a research issue in data provenance [59].

Data Curation

According to [60], data curation is "the active and on-going management of data through its life-cycle of interest and usefulness [...]. Curation activities and policies enable data discovery and retrieval, maintain data quality and add value, and provide for re-use over time". In the context of the Big Data value chain, A. Freitas et al. [52] defined the data curation activity as the process that "provides the methodological and technological data management support to address data quality issues maximizing the usability of data". These authors also remark the importance of the curation in the data integration and transformation tasks because it must facilitate these tasks to non-IT-experts. From this point of view, data curation is a key enabler of the self-service paradigm as described in Section 2.3.2.

In the same way, M. Stonebraker et al. [61] define this activity as "the act of discovering a data source of interest, cleaning and transforming the new data, semantically integrating it with other local data sources, and deduplicating the resulting composite". In this work, they remark the differences between traditional integration techniques such as ETL and data curation.

Although data curation mostly gives support to the data preparation step, it also takes part in other steps such as the data acquisition [61]. Data curation is then an activity which covers the whole life-cycle of data. It is concerned about data quality, the usefulness of data in the future, and the preservation of its value. Accordingly, C. Rusbridge [62] pointed at the need of an active and planned intervention in data. Regarding the self-service paradigm, data curation not only facilitates data transformation and integration tasks, but also the selection of relevant data sources which can enrich the data life-cycle. In conclusion, we consider data curation as an activity which gets benefited from the data quality and provenance tasks by combining the knowledge acquired from both of them in order to support other actvities of the Big Data Pipeline.

Data curation has become a major problem [63] in the Big Data context due to the volume and variety dimensions. The more data is generated and the more heterogeneous it is, the more quality, provenance, security and privacy issues appear.

Data Security

The data security activity consists of ensuring the integrity, availability, confidentiality and privacy of both the data and the whole Big Data Pipeline.

Data security is present in the whole Big Data Pipeline because it has influence in the rest of the activities [53]. The data security in Big Data environments can be studied from several points of view [64]: (i) the security of the underlying infrastructure; (ii) the access control policy; (iii) the real-time monitoring; and (iv) the privacy and confidentiality of the data.

The rise of the data security and privacy issues are motivated by the great expansion that the Big Data and IoT paradigms [65] [20] have experienced in recent years. The incorporation of business models based on Big Data and the irruption of IoT-based technologies make it necessary to guarantee data security and protect the privacy of the users. Recent regulatory frameworks related to data privacy also mark the importance of this activity in the pipeline [16] [53].

The implementation of security and privacy techniques in Big Data environments is challenged by the nature of Big Data, especially by the velocity, volume, and variety dimensions. As in the case of other activities, the huge amount of heterogeneous data that is generated at high speed challenge the control of the security of the data and all the activities which are part of the pipeline.

2.4 Big Data Architectures

Big Data architecture refers to the high-level components that enable the activities of a Big Data Pipeline. An architecture must be able to ingest and process large amounts of data, deal with both the intrinsic dimensions of Big Data [66], and the requirements of the use case, business process or context in which a data-driven solution is required. The *Big Data Architecture guide* created by Microsoft™[67] considers three basic use cases: (i) Big Data storage; (ii) data transformation and analysis; and (iii) capture, processing, and analysis of data streams in real-time.

Based on the literature, Big Data architectures might be characterized by the following aspects [68] [69]:

- **Scalability**. It is the capability to keep performance in case there is a growth in workloads. Systems can scale in two manners: horizontally and vertically. On the one hand, horizontal scaling [68] consists of distributing the workload across additional machines. On the other hand, vertical scaling consists of improving the features of a single machine. Big Data architectures must facilitate both kinds of scalabilities, especially the horizontal scaling, which is the most complex in terms of software implementation. The use case conditions the type of scalability to choose. For instance, iterative data analysis algorithms [68] work better in vertically-scaled systems because horizontally-scaled systems might overload network connections.

- **Fault tolerance**. It refers to the ability to continue operating correctly in case of any error [69]. Most Big Data architectures are required to be fault-tolerant and less error-prone by abstracting users from complex and repetitive tasks.

- **Latency**. The latency is the time the system takes for operations to be completed. As N. Marz et al. [69] say, most of the Big Data use cases require low latency. Getting low latency is not trivial, and it greatly influences the design of a Big Data architecture.

- **Extensibility**. It is the degree that the architecture is able to implement new functionalities and to support changes in the use cases.

- **Generalization**. It refers to the capability of the system to be adapted to new use cases.

Next subsections expose the most employed architectures in Industry to process large amounts of data. First, traditional Big Data architectures are introduced. Finally, two architectures for streaming data processing are described: the lambda and the kappa architectures.

2.4.1 Batch Processing Architectures

The first architectures for Big Data processing arose from the need to process large amounts of distributed raw data. Due to the volume, variety, and the velocity in which data was generated, traditional database systems were not able to offer proper scalability. In addition, software development in these systems became too complex [69] because programmers had to concern about the distributed nature of these systems (i.e., they were not abstracted from it).

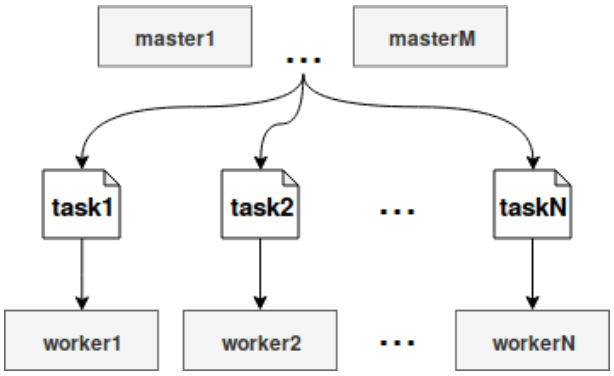


Fig. 2.3 Traditional Big Data architecture employed for batch processing.

In 2004, J. Dean et al. [70] proposed the *MapReduce* paradigm as one of the first solutions that enabled to solve the Big Data problem. It was based on a scalable, fault-tolerant, and robust architecture, abstracting developers from the distributed nature of the architecture. In this paradigm, the programmer had to specify the operations to be applied to the data. The software was executed autonomously when the programmer specified it. This is called *batch processing*.

From the logical point of view, Big Data architectures for batch processing are based on two types of components (i.e., master and worker) [70]. The aim of the architecture is to distribute the work among several workers, being the master component responsible for distributing the tasks. In a distributed system, there are several instances of these components. Figure 2.3 shows an example of a Big Data architecture for batch processing.

These architectures are scalable, fault tolerant and robust. On the one hand, their distributed nature makes it easier to scale horizontally. On the other hand, these architectures managed to abstract the developer from complex tasks related to the distribution and data processing, making it more fault tolerant and robust. The main limitation is that they tend to have very high latency. Therefore, these architectures are not appropriate to execute queries, make views or perform analysis in real-time scenarios.

2.4.2 Lambda Architecture

Traditional Big Data architectures succeeded in providing fault-tolerable and horizontal scalable solutions. Thanks to these architectures, solutions to deal with the volume, velocity and variety Big Data dimensions were developed.

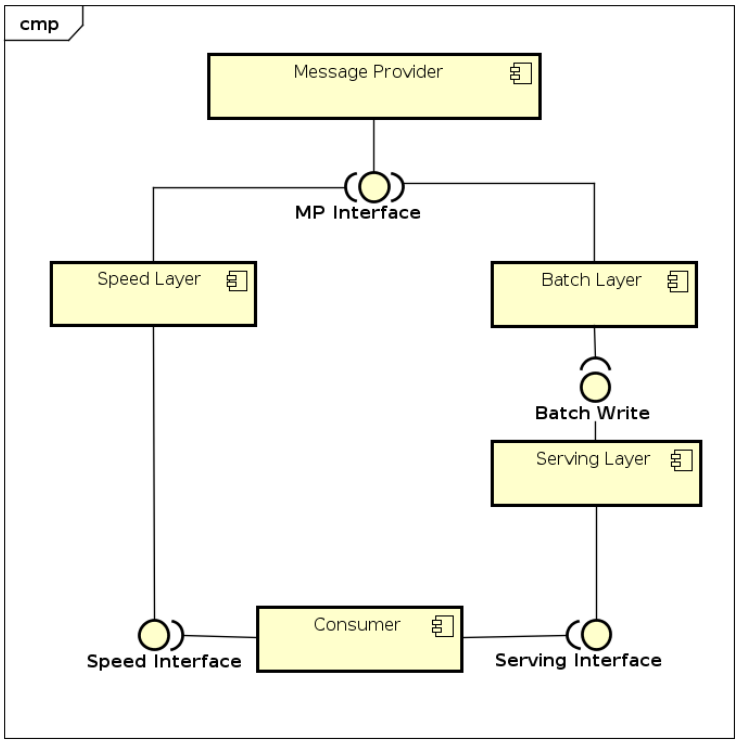


Fig. 2.4 Lambda architecture diagram.

On the other hand, these Big Data architectures were designed to perform batch processing. As explained above, this partially solves the problems of scalability and fault tolerance. Nevertheless, the latency of this type of architectures is very high. Some use cases require low-latency writing and/or reading operations. Hence, an architecture which requires real-time access to data is needed.

N. Marz et al. [69] proposed the lambda architecture in order to solve the problem of reading latency on data flow processing. It enables to make views and to perform queries to historical data, including the data that entered into the system at the moment in which the query was performed. In this way, it is possible to obtain updated results in real time.

The lambda architecture combines both batch-based and streaming-based data processing architectures. A diagram representing the lambda architecture is shown in Figure 2.4. The entry-point is a message provider that generates data in real time. The end-point of the architecture is a consumer whose objective is to make queries. Three layers are responsible for processing data.

- **Batch layer**. It receives and stores all messages sent by the message provider in real time. Its main function is to execute a job every certain period. The

objective of this job is to aggregate, structure, format, or perform other operations on the entire set of historical data in order to facilitate the subsequent processing. N. Marz et al. [69] highlighted the task of creating a fact-based model to facilitate subsequent indexing and querying tasks.

- **Serving layer**. It intends to create views from the results of batch layer jobs. These views enable to execute low-latency queries on the data generated as a result of the batch layer jobs. The serving layer is generally a database that must have four fundamental features [69]: (i) batch-writing support; (ii) scalability; (iii) low latency in random read operations; and (iv) fault tolerance.

- **Speed layer**. The speed layer receives data in real time and performs the same operations as in batch layer on demand. The dataset in the speed layer is smaller than in the batch layer because it only contains data generated since the last execution of the batch layer job.

One of the most significant features of the lambda architecture [69] is that data remains immutable during the whole process. The corresponding jobs are responsible for making aggregations and other operations that create new data from historical data.

The main advantage of the lambda architecture is the low latency. Thanks to the precomputed batch views, the speed layer does not have to process all the historical data on demand, but only those data that were generated since the last time the batch view was processed. This architecture facilitates the generalization and extensibility characteristics. On the one hand, it could be customized for multiple use cases that require real-time and low-latency access to data. On the other hand, it could be adapted [69] to new features or changes throughout the life-cycle of the system.

Although the lambda architecture significantly reduces latency problems, it presents a major drawback due to the complexity derived from maintaining and developing solutions based on this architecture. Two different architectures are required to be implemented in a lambda architecture, adding more complexity to tasks such as maintenance and debugging. Therefore, lambda architecture requires to develop the same software for at least two different technologies, making development tasks even more complex. J. Kreps [71] highlighted this problem, noting that developing software based on Big Data technologies is not a trivial task.

2.4.3 Kappa Architectures

As explained above, the lambda architecture solved latency problems by performing readings on both historical and real-time data. However, the deployment of two different architectures is required. J. Kreps [71] pointed out these problems and proposed an alternative solution.

A diagram representing this architecture is shown in Figure 2.5. The kappa architecture combine the speed and the batch layer into a single layer, which is called *streaming layer*. In this way, the complexity derived from having to implement and maintain two different architectures is eliminated. The consumer access the data through the serving layer, which stores the results sent by the streaming layer, which is responsible for processing the data in real time and delivering the results to the serving layer. Historical data would be reprocessed only if the request of the consumer changes. If a reprocessing of historical data is required, the streaming layer would instantiate a second instance of the processing engine [71].

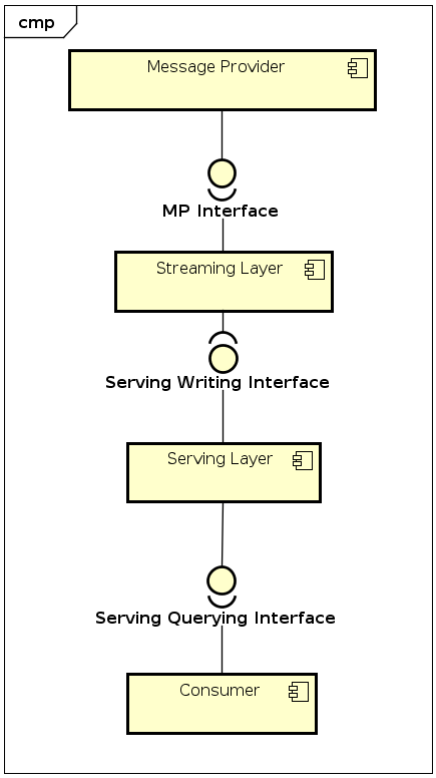


Fig. 2.5 Kappa architecture diagram.

In addition to providing access to data with low latency, the kappa architecture simplifies the design, facilitates its implementation, maintenance, and software

development. In conclusion, the kappa architecture provides low-latency access to data, and its design is simpler than the lambda architecture.

Regarding efficiency, it does not improve the lambda architecture, as remarked in the study conducted by V. Persico et al. [72]. In this study, the authors found that the lambda architecture slightly improves the kappa architecture in terms of execution time. Nevertheless, this study highlights that the performance kappa architecture improves better than the lambda architecture on vertically-scaling.

2.4.4 Internet of Things Architectures in Big Data Environments

The IoT concept refers to the connection of hundreds and thousands of devices to the Internet and to each other. The data generated in IoT contexts [73] can reach the three dimensions of Big Data: volume, because of the large amounts of data generated in short time by these devices; velocity, due to the speed with which the devices are able to capture and send the data; and variety, because of the format of the data may differ depending on the sensor or device that generates it. Therefore, IoT might challenge the data acquisition activity in the Big Data Pipeline. For this reason, some authors have put their efforts in studying IoT within the Big Data paradigm [73] [1] [74].

A. Taherkordi et al. [1] introduced the IoT Big Services concept. It refers to the integration of large amounts of data-centric services. It must meet the following requirements: (i) scalability; (ii) extensible; (iii) spatial and temporal context maintenance; and (iv) being based on network architecture and service model. The model of services proposed by these authors is based on tree structures, where the devices that capture the data are placed on the leaves, and are interconnected among them according to the spatial location or the type of service they offer. In this way, different devices can be connected, facilitating the management of these. Figure 2.6 summarizes this idea.

C. Cecchinel et al. [74] proposed an architecture based on a similar idea, but at lower level. It consists of the connection of different groups of sensors in *sensor boards*. They are connected to each other in *bridges*. These are responsible for aggregating data from the *sensor boards* and sending them to the cloud.

M. Marhani et al. [73] offered a higher level view of IoT architectures in Big Data environments. It consists of four layers: (i) IoT devices, which includes the devices responsible for capturing data; (ii) network devices, responsible for the interconnection between sensors and other IoT devices; (iii) IoT gateway, responsible

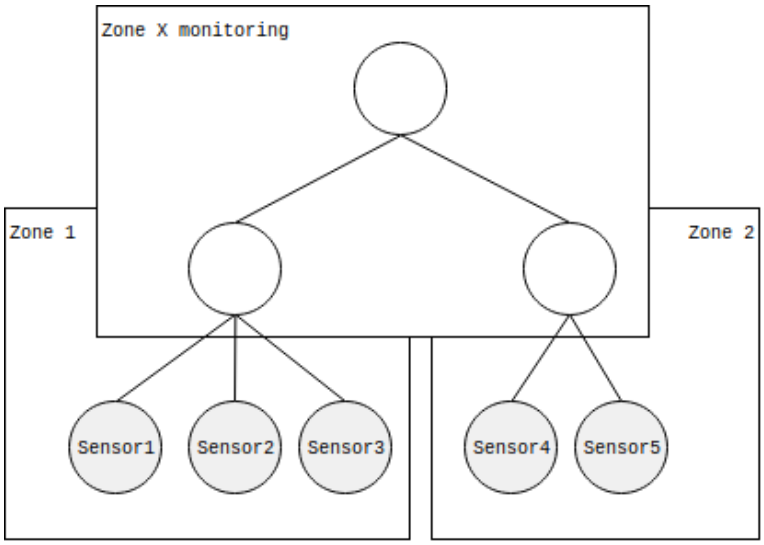


Fig. 2.6 A diagram representing the proposal of A. Taherkordi et al. [1]

for storing data in the cloud; and (iv) Big Data analytics, where data is processed to extract value.

The real-time data processing and the Big Data Pipeline that is applied in these cases do not differ from the exposed until now. A. Taherkordi et al. [1] specified the components of a Big Data architecture that would enable to process data from IoT devices. The proposed activities match with those that have been studied in Section 2.3 (e.g., data collection, data preparation, data storage, data analytics, and interpretation). Regarding real-time processing architectures, some authors in the literature use the lambda architecture to process the data captured by the IoT devices [75] [76].

To sum up, the integration of the IoT paradigm with Big Data is carried out in the data acquisition activity in the context of a Big Data Pipeline. The IoT paradigm requires the integration of numerous devices through networks. The design of this integration is very important to ensure the proper functioning and management of these networks. In this context, the communication protocols are essential [77].

2.5 Conclusions

The objective of this chapter is to research the foundations of Big Data and Big Data Pipelines based on an extensive literature review. Due to the fact that this field has become very disruptive, and the great interest that companies have on this, multitude of contributions about this field can be found in the literature. For this

reason, making a study of the state-of-the-art and establishing a clear and concise definition of the concepts related to Big Data and Big Data Pipelines is necessary.

Big Data has been defined as a field whose objective is to extract value from a type of data that fulfill three fundamental characteristics: volume, velocity and variety. These three characteristics (a.k.a., dimensions) have an enormous impact on the activities that are part of a Big Data Pipeline. The concept of Big Data Pipeline has been defined as the set of activities that enable the extraction of value from data. It is performed by means of the data analysis activity, the cornerstone of the value extraction process.

Other activities to highlight are the data provenance, data quality and data security. All of them have a great impact on the quality of the services provided by the Big Data Pipeline. Despite of this, there are not as many studies about these activities as on others, such as data preparation or data analysis. After the literature analysis, we can conclude that the most time-consuming activities are data acquisition on the one hand, and data preparation on the other. Regarding the latter, there are many very specific techniques to provide data with the appropriate format, facilitating the later analysis.

A study has been carried out about the main Big Data processing architectures that can be found in both Industry and in the literature. It has been found that there are solid solutions on the processing of data in real time, although some technical limitations have still to be overcome.

The work that has been carried out in this chapter draws the guidelines for the next one. Due to its relevance in the Big Data field, and the potential impact for organizations, it will be focused on the challenges and opportunities that arise from the following research topics: (i) data acquisition; (ii) data preparation; (iii) data analysis; (iv) data quality; (v) data provenance; (vi) data security; and (vii) Big Data Pipelines design.

Chapter 3

Research Findings

3.1 Introduction

In the previous chapter, the concept of Big Data and the principal activities involved in the Big Data Pipeline were described. However, most of these have not been fully adapted to the Big Data paradigm. For this reason, finding out the challenges, opportunities, and research issues which might be fundamental in the field of Big Data is needed. In order to find out the most relevant challenges, twenty research articles have been analyzed. All of them have been published between 2016 and 2018, due to the active an continuous evolution of this field.

This chapter is structured as follows. Section 3.2 draws the challenges and opportunities regarding some of the most challenging Big Data activities and research topics that can be found in the literature. Section 3.3 draws the most relevant research questions for each research topic. Finally, Section 3.4 concludes this chapter with a final analysis.

3.2 Challenges and opportunities

This section focuses on the challenges and opportunities of the most relevant activities of a Big Data Pipeline according to the study that has been carried out in Chapter 2. In Figure 3.1, the activities that are considered in this section are colored: (i) data acquisition; (ii) data preparation; (iii) data analysis; (iv) data quality; (v) data provenance; and (vi) data security. In addition, further challenges and opportunities regarding Big Data Pipelines in general are exposed.

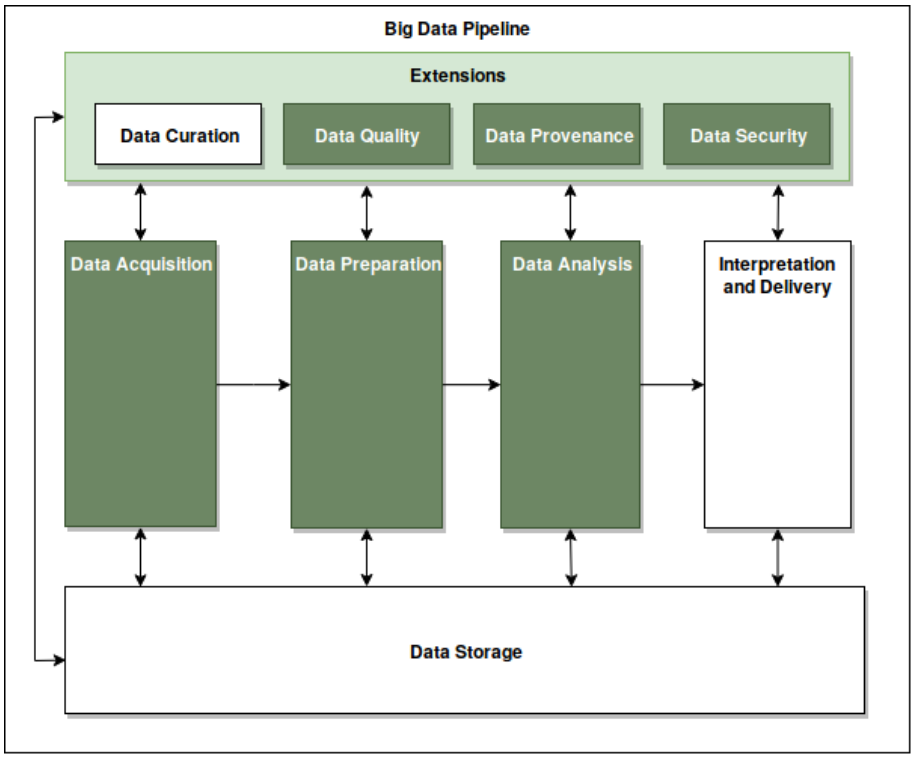


Fig. 3.1 Big Data Pipeline activities that are considered in this section.

3.2.1 Data Acquisition

This field has been tackled in previous works. However, there are still some challenges and limitations. Some of them are the integration of heterogeneous data sources, the integration between tools, and the application of operations during the ingestion of data, among others.

Data Acquisition from heterogeneous data sources and integration

The way in which the acquisition process is done, strongly depends on the data source. K. Lyko et al. [19] pointed out how this dependency hinders the data acquisition in a Big Data context, which in many use cases the connection between hundreds or thousands of data sources is needed. In the same way, C. Ardagna et al. [56] stated that due to the large amounts of data that are generated, and the increase of data sources for organizations, the data acquisition becomes a cumbersome in the Big Data Pipeline.

Opportunities for facilitating the data acquisition emerge, such as the development of a methodology to facilitate the integration of large amounts of heterogeneous data sources, including the development of protocols which facilitate the integration

of them. M. Shah [20] highlighted that the lack of standardized communication protocols for IoT devices also difficult the data acquisition. Although some efforts have been made in this direction, there is not a commonly accepted solution.

A. Taherkordi et al. [1] pointed out the challenge of the integration and management of spatially distributed devices. These authors developed a solution based on the concept of *Big Services*, explained in Section 2.4.4. Nonetheless, this paradigm requires more research efforts.

K. Lyko et al. [19] identified another challenge regarding the integration of the acquisition activity with the rest of Big Data Pipeline activities. They stated that this integration strongly depends on the technologies that are employed in the Big Data Pipeline, and the type of data processing (e.g., streaming or batch processing). C. Ardagna et al. [56] also stated this challenge.

In summary, the development of techniques to facilitate the integration of any type of acquisition technologies with data processing techniques is an opportunity.

Data Preparation Tasks During the Data Acquisition

Some authors have pointed out the challenge of performing data preparation tasks early in the data acquisition activity. K. Lyko et al. [19] distinguished some use cases where cleaning data before its processing is required. It might help to make a more efficient acquisition process, because low quality data would be filtered, contributing to the non-overload of the system. Other authors like O. Marcu et al. [78] also tackled the improvement of the data ingestion by applying some operators over the data stream.

The problem of performing data preparation in the acquisition stage is that data usually has not a specific format (usually it is raw data), and it might not have a clear semantic until it is put in context with other data. For this reason, authors like K. Lyko et al. [19] have noticed the difficulty of operating with data in this activity. Since performing some operations such as cleaning or aggregation would make more efficient ingestion process. Therefore, the development of techniques to facilitate the manipulation of data in the data acquisition activity is an opportunity.

Miscellaneous

Other challenges that can be found in the literature are listed below.

- K. Lyko et al. [19] et al. pointed out the difficulty of acquiring complex-format data such as image and video.

- M. Shah [20] highlighted the importance of guarantee the security and privacy of data sources in the data acquisition.
- P. Ceravolo et al. [16] mentioned the necessity of the development of mechanisms which enable the integration of data provenance in the acquisition activity.
- K. Lyko et al. [19] said that there is a wide variety of tools specialized in the collection and ingestion of data. These authors propose the development of a methodology to help the choice of technologies depending on the use case requirements.

3.2.2 Data Preparation

The data preparation activity is one of the most time-consuming tasks in the Big Data Pipeline. There are many activities and techniques that enable it. Challenges and opportunities regarding these techniques are discussed below.

Data Preparation and complex data structures

As the amount of data and data sources increase, the heterogeneity of the data tends to grow, and consequently, the complexity of these. In this new scenario, data with complex structures come into scene. These structures are commonly composed of nested schemata. Operations such as the integration and transformation of data with complex structure represent a challenge for the user [79] [35] [73].

J. Stefanowski et al. [79] indicated that as the complexity of the data increases, the techniques required to perform data preparation operations are more complex. Traditional techniques present significant limitations in this new context, because they are focused on the treatment of structured data with simple structures. The development of methodologies to facilitate these operations is an opportunity.

As indicated in 2.3.2, most data integration processes require the automatic discovery of the semantics of the data. However, the complexity of data makes these operations extremely difficult. Therefore, more research efforts are needed in this regard.

Data Integration

Data integration has been a concern during many years for IT technicians and researchers. The irruption of the Big Data paradigm brought new challenges to the

data integration area. M. Marjani et al. [73] highlighted the difficulty of integrating data due to heterogeneous schemata and the generation of data at high scale, and the need of real-time data integration. M. Shah et al. [20] and A. Siddiqa et al. [53] also emphasized these challenges. They focused in the importance of data integration in the context of *Business-to-Business integration*, where high-scale real-time integration is usually required. Both works propose the development of strategies, methodologies and tools to facilitate this type of integration as an opportunity.

In scenarios where large amounts of data sources must be integrated, the manual integration of data by the user is no longer feasible. Then, automatic data integration techniques are required [56]. However, it implies new major challenges:

- The automatic data integration require the detection of the semantic of each attribute in data. Establishing relationships between data generated from different sources is needed. It is performed by pairing attributes with the same semantic. M. Shah et al. [20] and P. Ceravolo et al. [16] pointed out the challenge of inferring the semantic of data taking into account its context.
- Once relationships between attributes have been devised, integration rules must be detected [16]. They enable the automatic transformation of data structures to homogenize them. However, in this kind of automatic process the uncertainty is a problem. Uncertainty is the degree in which the integration rule is not in accordance with the reality. Dealing with the uncertainty is a challenge in the Big Data context.
- Record linkage [56] is another problem in the automatic data integration. It consists of the alignment of records that represent the same instance, being an important challenge in the Big Data context.

Although there are proposals in the literature which concerns the automatic data integration [80] [16], most of them are focused on traditional system and do not scale well in Big Data systems. In addition, these proposals still require the participation of domain problem experts. Devising a flexible, adaptable and more automatic methodology to the Big Data context is a real opportunity [16].

Easing data preparation tasks

Data preparation tasks are significantly affected by the high-scale data generation [73]. P. Ceravolo et al. [16] highlighted as one of the most time-consuming tasks in

a Big Data Pipeline. If techniques were provided to abstract from low-level tasks, data preparation would not consume so much time and would be less error-prone. This idea is related to the self-service paradigm, explained in Section 2.3.2. Some approaches have been tackled in this direction [34]. Many other challenges remain, C. Ardagna et al. [56] pointed out that more research efforts on methodologies to facilitate the understanding of datsets by non-domain-experts are needed. It would provide users with semantic relationships between dataset, information on the data format, suggestions on possible integration, transformation or cleaning operations, and so on.

Data cleaning

Although data cleaning is a crucial task in Big Data scenarios, the volume, velocity, and variety difficult it. A. Siddiqa et al. [53] said that the development of data cleaning methods that fit the necessities of several use cases is not trivial. Each use case requires experts not only in the problem domain, but also in the different activities of a Big Data Pipeline. Another challenge is the real-time data cleaning, as M. Marjani et al. [73] stated. The problem here is that data cleaning requires the understanding of the semantics of data, and it is challenging in real-time scenarios, where the semantic and format of data might vary over time. The development of methodologies that enable data cleaning in many scenarios without the intervention of expert users is an important opportunity. Making these methodologies scalable and efficient is a major problem [7].

Miscellaneous

J. Stefanowski et al. [79] pointed out other problems that are related to the data preparation. These are listed below.

- The data structures and semantics might vary over time. Big Data system should be able to detect this evolution, and if it is convenient, the data preparation tasks and historic data should be adapted to these changes.
- The data preparation activity might include mechanisms to detect and perform integration with data that might improve the quality and the relevance of this data.

3.2.3 Data Analysis

Data analysis has been a research issue widely explored in both literature and Industry. Most of the challenges that data analysis is facing are derived from the necessity of real-time analysis at large scale. M. Marjani et al. [73] agree with this point of view, highlighting that the irruption IoT paradigm causes the need to read at high speed, including data with great heterogeneity.

According to this fact, Big Data challenged the capabilities of the first data analysis algorithms [79]. Aspects such as the velocity, volume, the variability in data format, the uncertainty, and real-time analysis are the stumbling block of current solutions. The development of algorithms with the following characteristics represent a challenge and are open research lines in the field of data science:

- Regarding the performance of the algorithms: the scalability, parallelization, and the capabilities of performing real-time analysis.
- Regarding other characteristics, being able to deal with: data provenance, privacy, data quality, and also being able to work with data with complex structures. Improving the transparency of data analysis is also a major challenge.

3.2.4 Data Quality

Among the extensions found in Figure 3.1, data quality in Big Data is becoming an essential aspect for organizations, because it leads them to obtain better results in the Big Data Pipeline [20]. Although during the last years some efforts have been devoted to this area, further research efforts are required in the Big Data context [15]. Some major challenges that can be found in the literature such as the uncertainty and completeness issues, the accuracy of the data, and the automation of data quality, are described below.

Uncertainty

P. Ceravolo et al. [16] and C. Ardagna et al. [56] stated that the uncertainty in data as a major problem especially in data integration. As mentioned in Section 3.2.2, automatic data integration might lead to uncertainty, and hence, generate low data quality. Overall, uncertainty comes when making assumptions about the data, and it might happen in any activity of the pipeline. Although there exists probabilistic models that allow assessing and detecting uncertainty, these are mainly focused

on traditional systems that do not scale well in Big Data systems. Therefore, the improvement and the adaptation of these approaches present a big challenge and an opportunity to deal with.

Accuracy

Assessing the accuracy of the data both before and after the analysis is a challenge. There are many factors that condition the data accuracy [56], such as the accuracy of the datasets that are being employed in the process, or the pipeline that is being employed to process the data. In this context, two challenges arise: (i) assessing the accuracy of the datasets that generated each piece of data; and (ii) being able to obtain a pipeline so that it optimizes the accuracy of the data analysis results. In both cases, data provenance is a key factor.

Automatic data quality assessment

The way in which data quality is assessed strongly depends on the nature of the data, its semantics, and therefore, the problem domain. On this basis, new methodologies [36] for the automatic data quality assessment are required. Thereby, challenges are: (i) as explained in Section 2.3.6, data quality depends on five dimensions that must be taken into account in the quality assessment; (ii) choosing and developing a set of metrics for each dimension; and (iii) being able to automatically suggest actions that would improve the data quality [9]. Although some efforts have been made in this direction [81], further research in this topic is needed.

Miscellaneous

Other challenges found in the literature [16], are listed below.

- Discovering and assessing data sources that might improve the quality of the data.
- The concept of "*quality-driven data access*", which consists of querying data by taking into account the data quality, is introduced. It might be helpful to find out data which fit some quality patterns. The development of methodologies and languages to support quality-driven queries is indeed a great opportunity.

3.2.5 Data Provenance

Data provenance is considered a key process to improve the credibility, integrity, and ultimately, the trustworthiness, the quality and the security of the data [55]. Although some efforts have been made in the context of traditional database systems, there are still many challenges and opportunities in the Big Data paradigm. Some major challenges are: (i) keeping track of the transformation applied to data; (ii) the size of the metadata; (iii) the interoperability; (iv) performing queries on metadata; and (v) the reproducibility.

Keeping track of the transformations applied to data

Keeping track of the data provenance in transformation operations is not a trivial task [55]. A good data provenance approach must be able to depict all the transformation performed over data and their lineage. In this regard, J. Stefanowski [79] et al. highlighted that techniques to support the data provenance in transformation operations have been developed for the MapReduce paradigm. However, there is a lack of solutions for other Big Data techniques.

The size of the metadata

A. Alkhalil et al. [55] stated that in the IoT context, devices generate large amounts of small data. In many cases, metadata might be larger than the data itself. Keeping track of these pieces of data is challenging. As J. Wang et al. [59] also highlighted, in order to get a detailed provenance track, the metadata usually tends to be higher than the data itself. In this regard, J. Stefanowski [79] et al. pointed out that methods to assess the relevance of provenance data have not been developed. These might be helpful to reduce the metadata size.

Interoperability

The heterogeneity of data sources is also a challenge. Data provenance should be able to operate among all of them [55]. In this regard, J. Wang et al. [59] highlighted that there is a lack of standardized models for data provenance in the Big Data context. Developing such a standard would be an opportunity.

Querying metadata

J. Stefanowski [79] et al. pointed out that querying and visualizing data provenance is challenging due to the size and complexity of the data about provenance, becoming especially complex when tracking the provenance in transformation operations. Nested and complex data structures also increase the difficulty of data provenance. In addition, there exist no standardized querying mechanisms to this end.

The necessity of methods to query metadata has been identified by other authors. A. Alkhalil et al. [55] established that indexing provenance and tools to perform queries on a particular context are required. J. Wang et al. [59] and I. Suriarachchi et al. [82] stated that there is a lack of flexible querying APIs to enable detailed provenance queries.

Reproducibility

P. Ceravolo et al. [16] and J. Wang et al. [59] pointed out the importance of being able to reproduce a Big Data Pipeline workflow. Data provenance might be the enabler for this task. Being able to track the workflow and reproduce it, would be an enabler for the development of techniques to (i) predict the performance of future workflows, and (ii) devise a workflow which maximizes the quality of the results [59].

Miscellaneous

Other challenges and opportunities related to data provenance that have been detected by other authors are listed bellow.

- C. Ardagna et al. [56] highlighted the importance of the data provenance in the trust assurance process. The implementation of techniques to ensure the trustworthiness and provenance is challenging in the context of Big Data analytics as-a-service.
- P. Ceravolo et al. [16] said that the development of data-provenance-oriented architectures is a challenge. These would facilitate the metadata management (e.g., the storage, discovery of metadata, etc.) and the consumption of the data provenance by other activities of the Big Data pipeline.
- A. Alkhalil et al. [55] said that in the IoT context, data provenance is challenged by security and privacy issues. Both of them must be guaranteed not only in the data itself, but in the meta-data.

- J. Wang et al. [59] highlighted that data provenance tasks implies high computational and network costs, especially in distributed environments. Hence, it tends to overload the primary Big Data activities.
- J. Wang et al. [59] also pointed out that including external metadata regarding the context of the environment in which the data is being processed (e.g., the configuration and performance of the system) is a challenge.

3.2.6 Data Security

Among the extensions found in Figure 3.1, data security is another important activity that is becoming of great interest of companies. Security issues such as vulnerabilities and privacy in Big Data systems are studied below.

Vulnerabilities in Big Data systems

Data security has become a major problem in Big Data, where vulnerabilities and privacy are one of the most relevant issues. Some authors, such as A. Siddiqa [53], A. Alkhalil et al. [55], M. Shah et al. [20] and M. Marjani et al. [73] stated that Big Data characteristics such as the generation of large amounts of data in real time, and the distributed nature of the Big Data architectures, might lead to important security issues. In addition, current security solutions are not efficient enough in this paradigm [15]. For instance, authenticating hundreds or thousands of devices is challenging as stated in [73].

Generally, most authors agree that the more entry-points the system has and the more heterogeneous it is, the more vulnerabilities the system will have. This is because controlling the security of such a complex system is not trivial. In addition to the difficulties associated to the management of data security in distributed systems, another major challenge is that security mechanisms might cause network overload [53].

C. Ardagna et al. [56] indicated that in the context of integration between different organizations, security and privacy play a compelling role. In this context, many risks arise regarding the integrity, privacy (due to the possibility of inferring data), confidentiality and availability. A. Siddiqa [53] established that the development of an efficient methodology or framework for security requirements assurance in the context of inter-organizational integration is needed. A. Siddiqa [53] also pointed out integrity and confidentiality issues especially in the context of inter-organizational integration, because un-trusted actors might have access to data.

Regarding the confidentiality, challenges appear because traditional encryption algorithms are not optimal enough in Big Data environments [55]. The development of efficient confidentiality mechanisms in Big Data is also an opportunity.

Privacy

A major challenge in Big Data security is ensuring privacy. A. Siddiqa [53] said that Big Data presents high privacy risks. One of them is that privacy might be violated by using inference.

In the context of the IoT, M. Marjani et al. [73] said that privacy becomes crucial because IoT devices usually are monitoring the user activity. It tends to produce distrust in users.

M. Shah et al. [20] highlighted the importance of anonymizing data sources. However, it might lead to conflicts with data provenance tasks, where identifying data sources is usually needed to provide a good provenance service.

Privacy politics are crucial to succeed in the privacy management. A. Siddiqa [53] and I. Lee [9] said that finding a balance between protecting privacy and the benefit that can be extracted from data analysis is a challenge. The development of frameworks that help to mitigate these risks while maintaining the potential benefit of analysis, and the development of metrics to measure privacy, are opportunities.

Miscellaneous

Other opportunities have been noticed by authors in the literature [10] [55] [9]. Some of them are listed below.

- Security politics models and prevention systems oriented towards the Big Data paradigm must be developed.
- The importance of the data provenance to provide support to Big Data security tasks.
- New technologies such as Blockchain might be useful for security management in Big Data environments. For instance, it might be employed in order to protect Big Data activities from attacks against data integrity.

3.2.7 Further Challenges and Opportunities

Further challenges and opportunities can be found in the literature [56]. Some of them are listed below.

- There is a vast amount of Big Data technologies in the market. Selecting a set of them to deploy a Big Data Pipeline might become overwhelming. Even more complicated is to find an optimal configuration for them. An opportunity could be the creation of a methodology for the selection of tools and configurations according to the needs of the pipeline.
- The design of a Big Data Pipeline which adapts to the needs of a use case can be complicated. In addition, verifying that the deployed Big Data system in a company correctly fulfills their requirements is a huge challenge. The development of methodologies to facilitate the design of Big Data Pipelines according to the needs of companies, and its assessment is an opportunity.
- The implementation of Big Data requires several highly qualified and highly specialized professional profiles. This increases the costs. The Big data-as-a-service paradigm, proposed by C. Ardagna et al. [56], tries to solve this aspect, but further efforts are required to achieve a correlation between the requirements of the use case and the Big Data service.
- There is a lack of international standards on Big Data management. Contributing to the development of such standards is an opportunity.

3.3 Research Questions

The previous section was intended to analyze the most recent challenges and opportunities which can be found in literature. In this section, a set of research questions are devised by taking into account some of these challenges and opportunities.

Data Preparation with Complex Data Structures

The challenges and opportunities regarding data preparation were depicted in Section 3.2.2. As stated in this section, the issues of processing of complex structures is one of the most prominent problems in Big Data. In this regard, we have focused the next research questions on the preparation of data with complex structures.

Research Question 1. *What are the problems of data preparation tasks with complex structures? How could these problems be solved in Big Data Pipelines?*

Research Question 2. *What mechanisms, methods, algorithms, and tools could be developed to make easier the preparation of data with complex structures in a Big Data Pipeline?*

Research Question 3. *How feasible is the automation of the data preparation with complex data structures in a Big Data Pipeline?*

Providing Data Quality, Provenance and Security in Big Data Pipelines

In Sections 3.2.4, 3.2.5 and 3.2.6, challenges and opportunities regarding the data provenance, quality and security were depicted. As mentioned there, data provenance might benefit quality and security. In addition, high-level modeling might be an enabler for these tasks. In this regard, the following research questions have risen.

Research Question 4. *How could data quality requirements be modeled from the high-level user objectives in a Big Data Pipeline? What are the mechanisms, methodologies, and tools that enable the application of such data quality requirements in the Big Data Pipeline independently of the underlying technology?*

Research Question 5. *How could security requirements be modeled from the high-level user objectives in a Big Data Pipeline? What are the mechanisms, methodologies, and tools that enable the application of such security requirements in the Big Data Pipeline independently of the underlying technology?*

Research Question 6. *How is data quality benefited from data provenance? What are the data quality dimensions which get more benefits from provenance?*

Research Question 7. *How can data provenance contribute to the security of Big Data systems?*

Research Question 8. *How can Blockchain technologies contribute to data provenance and security?*

Research Question 9. *How to develop data provenance standards for Big Data so that it does not depend on the Big Data Pipeline or the underlying technology?*

Research Question 10. *How to develop data quality, data security, and data provenance methods to face up the challenges arising from the data acquisition activity?*

Aligning Big Data Pipelines with Business Requirements

In Section 3.2.7, a set of challenges and opportunities about the Big Data Pipelines itself were depicted. Next, research questions which arise from those challenges and opportunities are shown.

Research Question 11. *Could a methodology be developed to help in the design of a Big Data Pipeline, the selection of technologies, and configurations aligned with the objectives and requirements of a company?*

Research Question 12. *What entry-barriers do companies face up with the deployment of a Big Data Pipeline? What mechanisms would enable companies to overcome these entry-barriers?*

3.4 Conclusions

The objective of this chapter is to determine the types of challenges and opportunities regarding the most relevant activities and aspects in the context of Big Data Pipelines. Three groups have been detected, aligned to: (i) data preparation; (ii) data provenance, quality and security; and (iii) business requirements. Although in many cases there are techniques to solve these challenges in traditional environments, it is necessary to adapt them to Big Data, since most of these challenges are related to the intrinsic characteristics of it. Some aspects such as quality and provenance have not become relevant until the appearance and consolidation of Big Data in companies.

These fields require much more research efforts and might help significantly improve the process of extracting value from the data. Nonetheless, not all of these challenges can be carried out in this master thesis. For this reason, the proposal presented Chapter 4 is focused on the research question 2, where the efforts are centered on the preparation of data with complex structures.

Chapter 4

Proposal

4.1 Introduction

In previous chapters, the state-of-the art of Big Data and Big Data Pipelines has been studied. In this chapter, a solution to one of the research questions that were formulated in Chapter 3 is proposed. The question number 2 has been chosen. It is: *What mechanisms, methods, algorithms, and tools could be developed to make easier the preparation of data with complex structures in a Big Data Pipeline?*.

As explained above, the transformation of data with complex structures in Big Data environments is a problem that hinders data preparation tasks in general, and data integration and transformation in particular. The proposal is a framework for performing transformations of complex data structures, and a Domain-Specific Language (hereinafter, DSL) so that end-users can perform transformation operations on data with nested structures (e.g., arrays and nested schemata). As P. Ceravolo et al. [16] said, DSLs based on an existing programming language are preferred by data scientists, who are the professionals who normally have to deal with data preparation tasks. For this reason, this proposal is based on a DSL implemented on the Scala programming language, which is one of the most used in the Big Data field. In addition, the proposed DSL is tested in a Big Data environment by using Apache Spark.

The rest of this chapter is structured as follows. Section 4.2 depicts the case study that will be used to guide the proposal. Section 4.3 describes the transformation framework that has been proposed. First, some concepts regarding it are defined. The model of the framework is then described. Next, the DSL definition is shown. Finally, the transformations that solve the case study are implemented. A set of

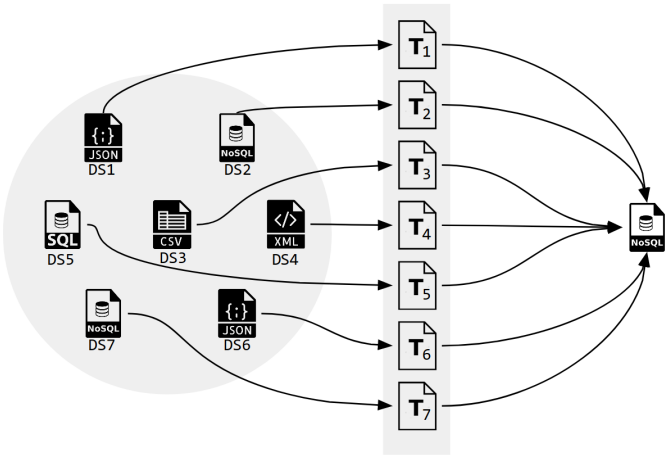


Fig. 4.1 Data preparation scenario.

benchmarks to test the proposal are presented in Section 4.4. This chapter concludes with Section 4.5, where conclusions and future work are drawn.

4.2 Case Study

The example is a real-world scenario based on the data transformation provided by seven electricity companies that sell energy for private customers in Spain. The electricity wholesales describe consumption data in different formats and using different frequency of meter reading, depending on factors such as the distributor or the tariff hired by each customer. These various formats need to be uniform in order to facilitate the data processing and analysis. An example of use case is the detection of behavior patterns or to look for the best tariff for each customer [83]. However, each electricity provider offers information using different nested schemata, depending on the number of months included in the meter reading, number of days, types of tariff, etc. Therefore, all of these heterogeneous schemata need to be transformed into a unified one. Figure 4.1 illustrates the scenario where several data sources must be conciliated into a unified format accessible to the final user. According to the companies involved in this use case, the solutions today available in the market are not able to tackle these transformations.

As mentioned above, the provided information does not follow the same schema, but they generally share a customer ID, a tariff identifier, the contracted power for each daily billing period, and a list of consumption over a period (e.g., twelve months). Each consumption period keeps information on the start and end date for

that period, and the power consumption for each daily billing period. Figure 4.2 shows a possible input schema for the data of the example and its relationships with the target schema.

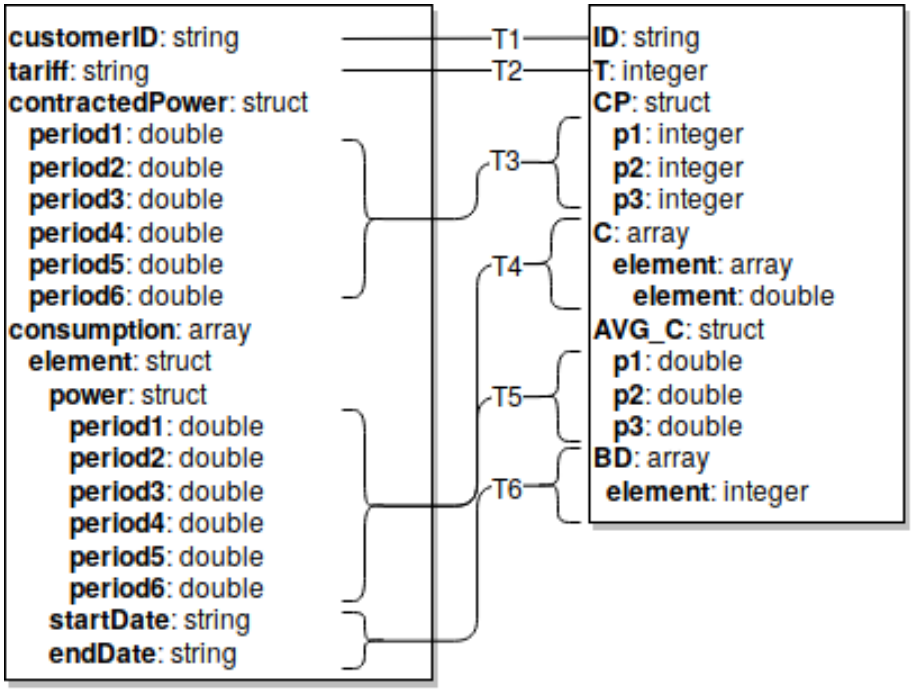


Fig. 4.2 Transformations to be performed. Left: source schema. Right: target schema.

4.2.1 Source Schema Description

The source schema is composed of basic and nested attributes. The description of the dataset attributes is given as follows:

- **customerID**: string which identifies a unique customer supply point.
- **tariff**: rates hired by the customer. It is composed of a string that varies depending on the company that distributes the electricity.
- **contractedPower**: the hired power for each daily billing period. It is a data structure with six decimal numeric attributes, each one representing the contracted power for a daily billing period.
- **consumption**: power consumption over a period, such as twelve months or more. It is an array of data structures. Each element represents a period, and it includes the following information:

- **power**: the power consumption for each daily billing period. It has the same structure as **contractedPower**.
- **startDate**: start date for the billing period.
- **endDate**: end date for the billing period.

4.2.2 Transformations and Target Schema

In order to reach the target schema, several transformations must be applied for each electricity supplier. In the case study presented above, six transformations are needed as depicted in Figure 4.1:

T1. *customerID* must be renamed to *ID*. No further transformations are required.

T2. *tariff* is transformed into an integer value *T*. The value must be transformed since in the source dataset the same tariff can be represented by several strings. These strings are unified in a unique number for each type of tariff.

T3. *contractedPower* is transformed into *CP*. Unlike the source data structure, it is composed of three integer attributes: *p1*, *p2* and *p3*. Each one is calculated as follows: Let $period_i, i \in [1,6]$ be each daily billing period. The power used to calculate the electricity price invoiced to each customer, in accordance with the defined rules by the government, is calculated in this way:

$$\forall p_k, k \in [1,3] : p_k = max(period_k, period_{k+3}) \tag{4.1}$$

T4. *consumption* is transformed into a matrix (*C*), whose rows have three elements that are calculated from the source attributes of *power* stated in the same way that *CP*.

T5. In the target schema, *AVG_C* is a data structure with three elements: *p1*, *p2* and *p3*. Each one is the average of the resulting value of the calculation explained in the third transformation applied to each element in the *power* structure inside the *consumption* attribute.

T6. In the target schema, *BD* is an array of integer values. Each value is the number of days of the corresponding period, calculated from the attributes *startDate* and *endDate* in the *consumption* array.

4.3 Data Transformation Framework

In this section, the framework that enables the transformation of data with complex structures is presented. First of all, the basic concepts that are related to the transformation of data are defined. Finally, the components of this framework that enable the transformation operators are modeled.

4.3.1 Related Concepts

Next, the concepts *Data Schema*, *data type*, *Transformation Function*, *Source Schema* and *Target Schema* are defined. These concepts will facilitate the understanding of the framework that is modeled in Section 4.3.2.

Definition. *A Data Schema (DS) is a set of attributes,* $\{a_1 : t_1, a_2 : t_2, \ldots, a_n : t_n\}$ *identified by a name* (a_i) *and a data type* (t_i).

Regarding the data type (t_i), two categories of data types have been identified. These categories are detailed as follows:

- **Simple Type**. It is a data type which represents a single value:
  - **Numeric**. It represents a numeric data type (i.e., Integer, Long, Float, and Double).
  - **String**. It is a sequence of characters.
  - **Boolean**. It is a two-valued data type which represents the truth values.
  - **Date**. It is a set of characters with a specific format that represents an instant of time.
- **Complex type**. It is a composite data type that can be:
  - **Array**. It is a collection of typed attributes identified with a unique numeric index.
  - **Struct**. It is a data type composed of a set of attributes, each one identified by a unique name.

Definition. *A Transformation Function,* t_x*, is a function that receives an attribute,* a_{input}*, and returns an attribute,* a_{output}*, as a result of applying an operation which modifies the value of* a_{input}.

$$t_x : a_{input} \rightarrow a_{output} \tag{4.2}$$

Definition. *Let Source Schema (S_{Source}) be a DS of which attributes are used as a_{input} of a t_x.*

Definition. *Let Target Schema (S_{Target}) be a DS composed of a set of attributes that are the a_{output} of a set of t_x.*

In order to transform a Source Schema (S_{Source}) into a Target Schema, (S_{Target}), a set of transformation functions are required. A framework has been modeled to support such transformation functions. It is presented in the next section.

4.3.2 Framework Modeling

First of all, the data types have been modeled. Figure 4.3 shows the data types modeling in accordance with what has been exposed in Section 4.3.1.

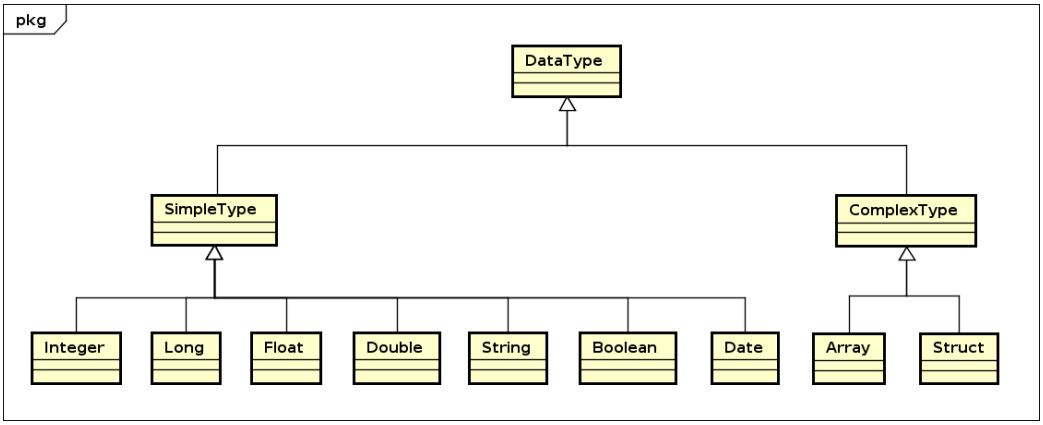


Fig. 4.3 Data Types that are employed in this proposal.

The framework has been designed according to the composite design pattern [84]. In short, this pattern enables to build complex objects by using simpler ones. It means that an object could be composed of nested objects. Figure 4.4 depicts a schema of this pattern. As can bee seen, the classes *Composite1* and *Composite2* are composed of a set of *Components*, which can be *Composite1*, *Composite2*, or *Leaf*. The latter is called *Leaf* because it is not compounded by any other *Component*. In this pattern, the instances of objects could be represented as a tree structure.

Figure 4.5 depicts the UML diagram of the transformation framework. As mentioned above, the instances of this model can be represented as a tree structure. In

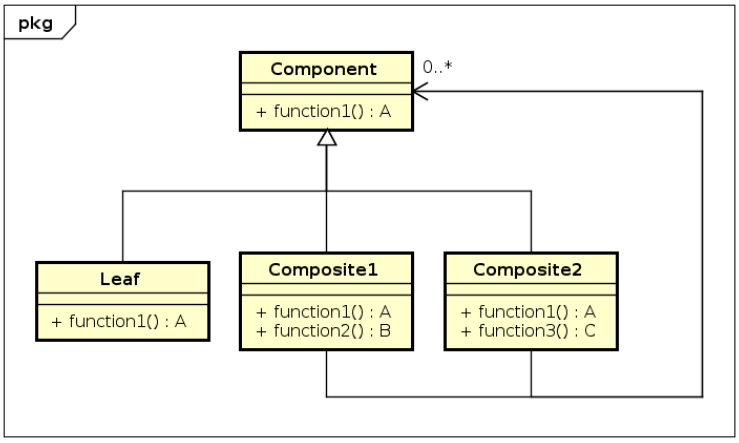


Fig. 4.4 The composite design pattern.

this structure, leafs are operations that access the attributes, and internal nodes are intended to transform or create new structures. To better understand it, Figure 4.6 shows an instance of the transformation T1 exposed in Section 4.2.2, and Figure 4.7 shows its tree representation.

In this model, the *Component* is the *Evaluable* interface. An *Evaluable* represents an expression whose main objective is to perform transformation functions on attributes. Two methods can be applied over every *Evaluable* expression (hereinafter, expression): *getValue* and *getDataType*.

- **getValue**. It receives an attribute, and returns another attribute as a result of applying a transformation to it.
- **getDataType**. It receives a data type, and returns the data type as a result of applying a transformation to it.

These are intended to be the entry-point of the framework. The way these functions work depends on the *Leaf* or the *Composite* components. Next, the *Leafs* of the transformation framework model are listed.

- **Select**. It is meant to select the attribute whose name matches the string *name* from an attribute of type *Struct*.
- **Index**. It is meant to select the attribute whose position matches the integer *index* from an attribute of type *Array*.

Lastly, the *Composites* of the transformation framework model are listed.

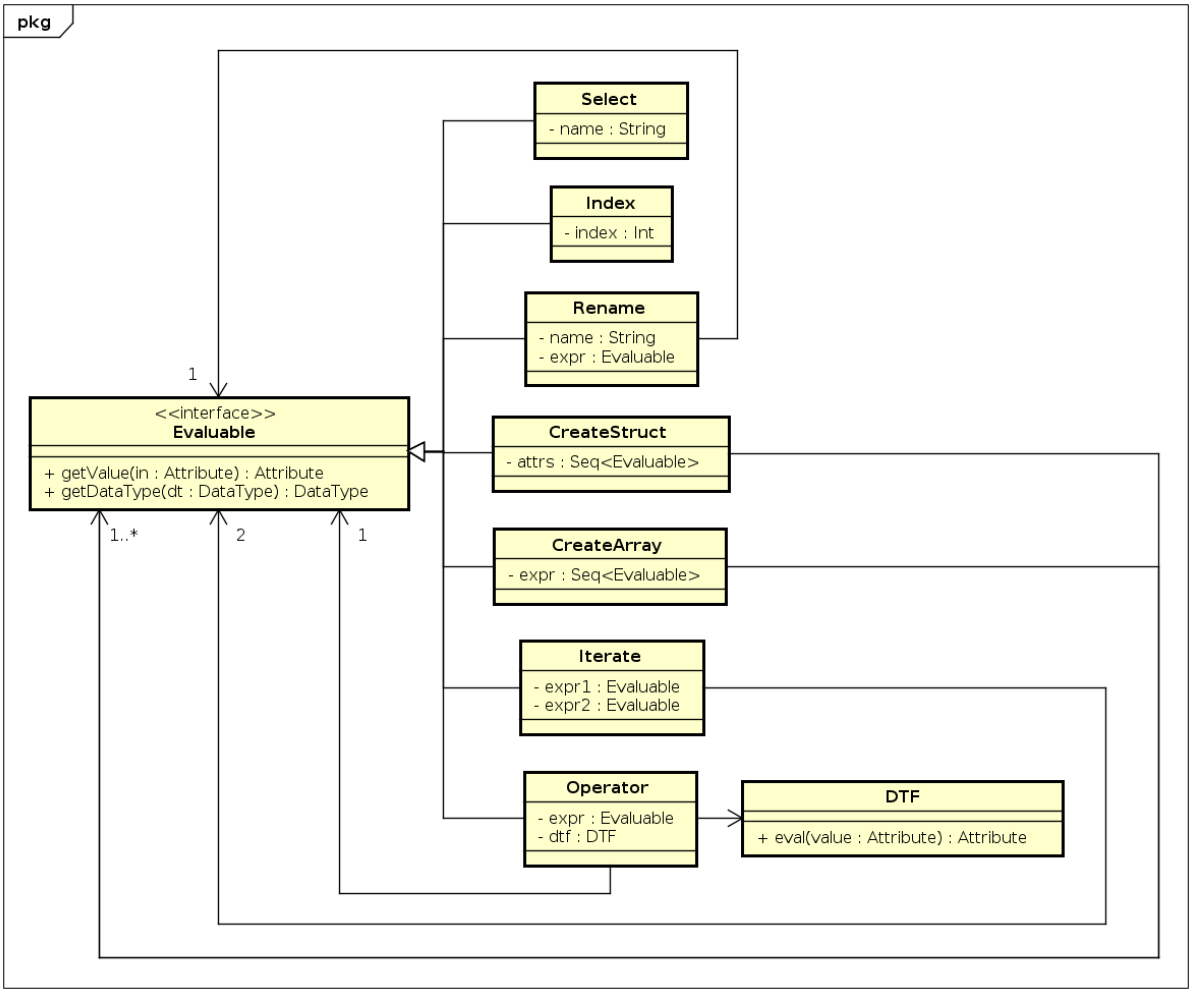


Fig. 4.5 UML model of the proposed transformation framework.

```
new Rename("ID", new Select("customerID"))
```

Fig. 4.6 Instance of the model to perform the transformation T1. Code representation.

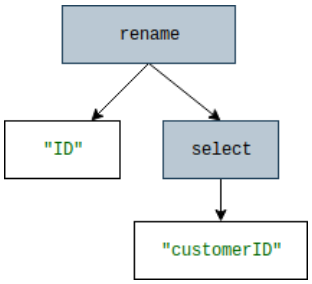


Fig. 4.7 Instance of the model to perform the transformation T1. Tree representation.

- **Rename**. It is meant to transform an *Evaluable* expression (hereinafter, expression) which return an attribute of any type by replacing its name by the string *name*.
- **CreateStruct**. It is meant to create an attribute of type *Struct* from a set of expressions *attrs*.
- **CreateArray**. It is meant to create an attribute of type *Array* from a set of expressions *attrs*.
- **Iterate**. It is meant to create an attribute of type *Array* as a result of iterating over an expression which returns an attribute of type *Array* (*expr1*). An expression (*expr2*) is applied to each element in that *Array*.
- **Operator**. It is meant to transform an expression by applying a *Data Transformation Function* (hereinafter, *DTF*).

DTFs are intended to apply a transformation function to an expression. These enable users to perform advanced transformations on attributes of any data type. Next, a set of them are grouped and listed. Additionally, users might define their own *DTFs*.

- **Reduction**. The following *DTFs* are meant to receive an expression which returns an attribute of type *Array*, and return the maximum, minimum, average, and the sum of all values of such *Array*, respectively: *max*, *min*, *avg* and *sum*.
- **Transformation of Data Types**. The following *DTFs* are meant to receive an expression which returns an attribute of type *Simple*, and return a value of type *Integer*, *Long*, *Float*, *Double*, *String*, *Boolean* and *Date*, respectively: *toInt*, *toLong*, *toFloat*, *toDouble*, *toString*, *toBoolean* and *toDate*.
- **Data Modification**. The following *DTFs* are meant to modify the value and data type of the attribute returned by an expression:
  - *scale*. This *DTF* works with expressions which return numeric attributes. It receives an *Integer* value *n* and creates a *Numeric* attribute by multiplying by *n* the attribute returned by the expression.
  - *translate*. It receives a *key-value* data structure and replaces the value of the attribute returned by the expression accordingly to the *key-value* data structure.

– *repeat*. It receives an *Integer* value *n* and creates an *Array* attribute by repeating the attribute returned by the expression *n* times.

4.3.3 Domain-Specific Language

Due to the high complexity of the transformations, a versatile user-friendly DSL has been defined in order to enable users to perform transformation operations on complex data structures. The syntax and grammar here proposed are intended to be intuitive and concise so that the learning curve is not very steep. The syntax of the grammar is given below by means of Extended Backus–Naur form notation (hereinafter, EBNF notation) [85].

Syntax

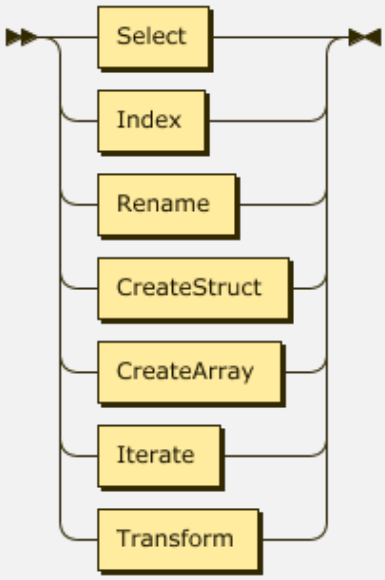
Syntax is the entry-point to the DSL. It is given by an *Expression*.

```
Syntax ::= Expression
```

Expression

Expression might be one of the following: *Select*, *Index*, *Rename*, *CreateStruct*, *CreateArray*, *Iterate* or *Transform*.

```
Expression ::= Select
      | Index
      | Rename
      | CreateStruct
      | CreateArray
      | Iterate
      | Transform
```



Select

Select is meant to be the syntax employed to select an attribute in a data structure. The selection is performed by specifying the name of the attribute.

```
Select ::= StringLiteral
```

Index

Index is meant to be the syntax employed to select an attribute in an array. The position of the attribute to select is given by its digits.

```
Index ::= Digits
```

Rename

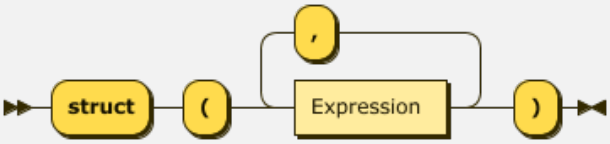
Rename is intended to enable to change the name of an attribute. The new name of the attribute is given by a string literal. The character "<" is employed as an assignment operator. Then, at the right of the operator, the *Expression* which is assigned to the name specified before is given.

```
Rename ::= StringLiteral '<' Expression
```

CreateStruct

CreateStruct enables to create an attribute of type *Struct*. In order to create it, the reserved word "struct" must be employed. The expression or expressions that will form the attribute of type *Struct* must be specified in parentheses separated by commas.

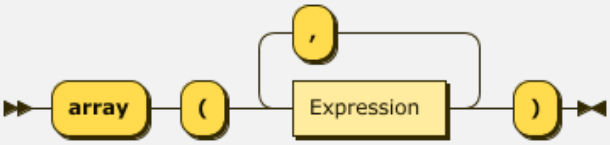
```
CreateStruct ::=
  'struct' '(' Expression ( ',' <Expression> )* ')'
```



CreateArray

CreateArray enables to create an attribute of type *Array*. In order to create it, the reserved word "array" must be employed. The expression or expressions that will form the attribute of type *Array* must be specified in parentheses separated by commas.

```
CreateArray ::=
  'array' '(' Expression ( ',' <Expression> )* ')'
```



Iterate

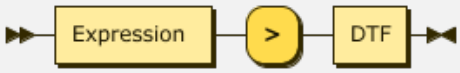
Iterate enables to perform an operation over an attribute of type *Array*. First, the expression to move through must be specified. Next, the reserved word "iterate" is employed. Finally, the expression that specifies the operation to apply to all the elements in the first expression must be specified.

```
Iterate  ::= Expression 'iterate' Expresion
```

Transform

Transform enables to apply a transformation function to an expression. First, the expression to which the transformation will be applied is placed. Next, the reserved word ">" is employed. Finally, the transformation (*DTF*) to apply is indicated.

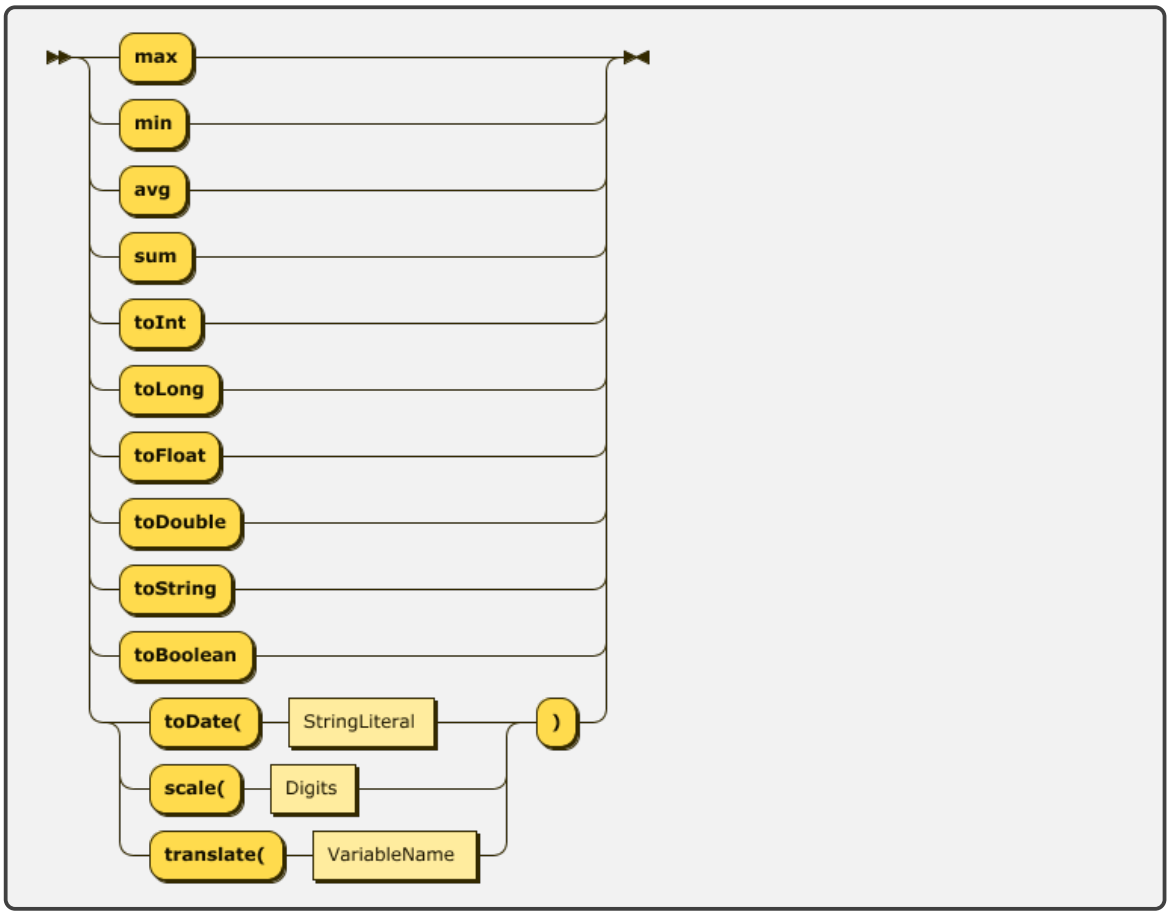
```
Transform ::= Expression '>' DTF
```



DTF

DTF represents all the transformation functions defined in 4.3.2.

```
DTF ::= 'max'
  | 'min'
  | 'avg'
  | 'sum'
  | 'toInt'
  | 'toLong'
  | 'toFloat'
  | 'toDouble'
  | 'toString'
  | 'toBoolean'
  | ( 'toDate(' StringLiteral ')' |
      'scale(' Digits ')' |
      'translate(' VariableName ')'
    )
```



4.3.4 Case study Transformations

Once that both the transformation framework and the DSL have been presented, the transformations that enable to perform the operations specified in 4.2.2 are implemented. Since the DSL is based on the Scala programming language, its syntax is used in some cases. It is assumed that the reader knows the grammar and data structures of Scala, ergo, only those aspects related to the DSL defined above will be explained.

T1. *customerID* is renamed to *ID* by using the " $<$ " operator as follows.

```
"ID" < "customerID"
```

T2. *tariff* is transformed according to the equivalences defined in a Scala *Map* object that describes the mapping between *tariff* names. For example, "*TA*" to 1, "*TB*" to 2 and so on. The " $<$ " operator is used to assign the resulting

value to the new attribute "*T*". The transformation, "*translate*", is applied by using the " > " operator.

```
"T" < "tariff" > translate(dictionary)
```

T3. An *Struct* attribute composed of three attributes is created by using the "*struct*" operator. Each element of this structure is generated by using the " < " operator to rename the output, the "*array*" and " > " operators, which enable to apply the "*max*" and "*toInt*" *DTFs* by specifying the fields to transform inside the "*array*" declaration.

```
"CP" < struct (
  "p1" < array("contractedPower.period1", "contractedPower.period4")
         > max > toInt,
  "p2" < array("contractedPower.period2", "contractedPower.period5")
         > max > toInt,
  "p3" < array("contractedPower.period3", "contractedPower.period6")
         > max > toInt
)
```

T4. In this case, an array of arrays is created by using the "*iterate*" operator. In this case each tuple of *consumption* is analyzed, obtaining a new array with three attributes by using the "*array*" operator. Each value of the new array is obtained by using the "*max*" *DTF* by means of the " > " operator.

```
"C" < ("consumption" iterate (
    (1 to 3) map(i => array(s"power.period$i", s"power.period${i+3}") > max)
  )
)
```

T5. The *AVG_C* attribute is similar to the transformations explained above. An structure *AVG_C* is created with three attributes by using the "*struct*" operator. Each value of the attributes is obtained by applying the "*avg*" *DTF* over the maximum of consumptions.

```
"AVG_C" < struct (
  (1 to 3) map(i =>
    s"p$i" < array(
      "consumption" iterate array(s"power.period$i",s"power.period${i+3}") > max
    ) > avg
  )
)
```

T6. The *BD* attribute is obtained similarly. The main different is the use of two *DTFs* that have not been previously employed: "*avg*" and "*toDate*". In addition, a custom *DTF* is employed: *daysBetweenDates*, which calculates the number of days between two dates.

```
"BD" < array ("consumption" iterate array(
    "endDate" > toDate("dd/MM/yyyy"),
    "startDate" > toDate("dd/MM/yyyy")
  ) > daysBetweenDates
)
```

4.4 Benchmarking

A set of tests has been devised in order to evaluate and check the performance in a Big Data environment. First of all, the Big Data architecture used to perform the tests is presented. Afterward, the datasets, tests, and benchmarks are described in the evaluation design. Finally, the results are drawn and discussed.

4.4.1 Architecture and Implementation

The architecture employed to perform the benchmark is based on a cluster managed by Mesosphere DC/OS (hereinafter DC/OS) [86]. DC/OS is an operating system based on Apache Mesos [87], which enables the execution of technologies for simultaneous data processing. In this case, an Apache Spark cluster has been deployed [88] [89] together with Spark History Server, that enables to extract execution metrics of the Apache Spark applications.

Regarding the infrastructure, it consists of a DC/OS *master* node, responsible for managing the cluster resources and assign them to services, and nine *agents*, responsible for managing the services. The instance of Spark includes a *driver* and nine *executors*. The architecture also includes a node with HDFS [90] to store the datasets and a MongoDB [91] database for storing the execution results. Regarding the computational characteristics, the cluster can reach fifty-two cores between 2 and 2,6 GHz for each and 136 gigabytes of RAM in global. Figure 4.8 depicts the infrastructure as well as the computational characteristics of the cluster. Summing up, the cluster can reach fifty-two cores and 136 gigabytes of principal memory in global.

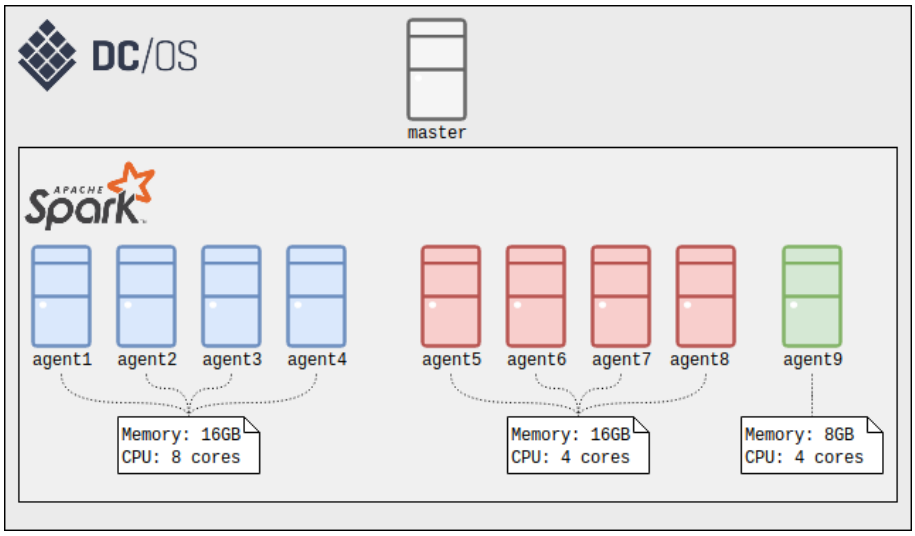


Fig. 4.8 Architecture of the cluster.

Table 4.1 Dataset and Benchmarks for the evaluation

Dataset ID	Criteria	Size (MB)	Benchmark
*D*1	A	4,119.4	1
*D*2	A	6,178.4	1
*D*3	A	8,239.9	1
*D*4	A	10,299.9	1
*D*5	B	3,659.8	2
*D*6	B	5,258.9	2
*D*7	B	6,859.4	2
*D*8	B	8,459.2	2
*D*9	B	10,060.3	2

4.4.2 Evaluation Design

The dataset used to plan the benchmarks has the same data schema presented in the case study with approximately more than five million tuples and a size of 2,1 GB. In order to test the scalability of the proposal, nine additional datasets datasets have been created based on two different criteria: (A) four new datasets by increasing the number of tuples; and (B) five new datasets by increasing the size of each tuple (i.e., by increasing the size of the columns), for instance, by duplicating the number of elements in the *consumption* array attribute. Table 4.1 summarizes the datasets which have been synthetically created by using these two criteria.

Ten test cases have been defined, each of them being executed one hundred times. These tests cases have been classified into two groups of benchmarks: (i) *Benchmark 1* where these test cases are intended to check the performance when the dataset size increases by the criteria A; (ii) the *Benchmark 2* where these test cases are intended to

check the performance when the dataset size increases by the criteria B. In each test case, all transformations described in Section 4.2 have been applied for each tuple of the dataset.

Both benchmarks have been developed by using an Apache Spark application. The application consists of two main stages. The former reads the dataset from HDFS and infers its schema, and the latter distributes the tuples across the cluster, applies the transformations and finally stores the results in MongoDB. As for performance metrics, both the Elapsed Real Time (*ERT*) and the *CPU Time* of the second stage have been measured in each test case. The *ERT* is the execution time since the stage corresponding to the application of the transformations is launched until it ends. On the other hand, the *CPU Time* is a time accumulator that includes the time the tasks related to the transformations spent on the CPU. For each test case, the average value of one hundred executions will be considered.

4.4.3 Evaluation Results

The results for both benchmarks have been depicted in Figure 4.9. A trend line has been included in the charts in order to highlight the tendency of the results. The least-squares fitting method has been employed in order to calculate the trend lines.

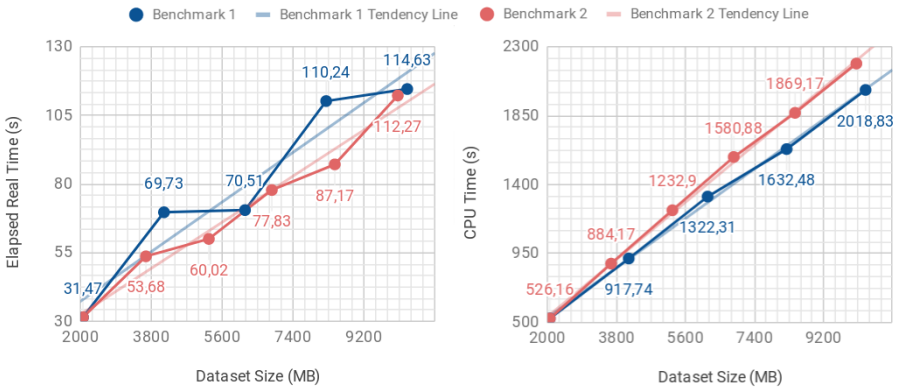


Fig. 4.9 Comparison between Elapsed Real Time (left) and CPU Time (right).

The chart on the left side shows the *ERT* comparison between both benchmarks. It can be seen that the *ERT* tends to be higher in the case of the *Benchmark 1*. It means that for datasets with the same size, the *ERT* is greater for datasets with more tuples to process than for datasets with less but more complex tuples. This is because the distribution cost is higher when processing a higher amount of tuples.

The right-side chart shows the *CPU Time* comparison between both benchmarks. Unlike in the case of the *ERT*, the trend line of the *Benchmark 2* tends to be higher than the *Benchmark 1*.

The trend line equations are $y = 0.18x + 174$ for *Benchmark 1* whilst for *Benchmark 2* is $y = 0.21x + 122$. In fact, the complexity of the dataset used for the *Benchmark 2* is higher as its tuples are larger than in *Benchmark 1*, consuming each one more CPU time. Despite this fact, there is only a 15% of the difference between the slope of the trend lines, being both lines under linear.

As a conclusion, it is possible to confirm that the proposal scales in regard to the dataset complexity because the increment on the complexity on processing nested structures and hence the transformations to apply, only suppose a 15% in regard to processing a dataset with less-complex structures.

4.5 Conclusions

Data transformation is a crucial stage of data analytics that is not fully addressed by Big Data technologies. For this reason, in this work, a concise and flexible DSL supporting a set of complex transformation functions is presented. The proposed solution has been developed on a Big Data framework based on Apache Spark and Apache Mesos infrastructure. Then, it has been evaluated with several datasets transformed according to a real-world case study. The conclusions show that the elapse time is more affected by the number of tuples while the CPU time is affected by the size of the tuples. Moreover, this proposal demonstrated to scale under linearly to the size of the dataset.

There are some limitations that will be faced in the future. For instance, the framework might include conditional operators in order to allow users to filter arrays. The inclusion of additional *DTFs* and the development of connectors to integrate this framework with further Big Data environments are examples of improvements which might be implemented in the future. Finally, the Big Data context is facing an important challenge regarding the veracity and quality of the processed data. This proposal does not implement any methodology to assess neither the quality of the dataset to be transformed nor the quality of the resulting dataset. Nevertheless, these aspects are quite relevant and need to be analyzed in depth.

Chapter 5

Conclusion and Future Work

5.1 Conclusion

The objective of this work has been to carry out a study on the state-of-the-art of Big Data and the activities that enable it. This study is motivated by the great interest this field has in the literature and Industry, and its continuous evolution, mainly due to the irruption of the Big Data paradigm in organizations that see it as a great opportunity to improve their business processes.

The contextualization of concepts and activities related to Big Data and Big Data Pipelines has been an enabler to the better understanding of the Big Data Pipeline as a whole. Big Data was defined as a type of data which fulfills three characteristics: volume, velocity and variety. As demonstrated in this work, these characteristics (a.k.a dimensions) strongly condition the activities associated with the value extraction process: Big Data Pipeline. In this regard, a global schema which covers the activities related to Big Data Pipelines was proposed. It was defined according to studies and proposals that were carried out by other authors in the literature.

Once Big Data and Big Data Pipelines were contextualized, the limitations, challenges, opportunities, and possible research lines were studied. This part of the study concluded that most of the challenges that the most relevant Big Data activities face up are related to the intrinsic characteristics of Big Data. Many traditional techniques that deal with data must be adapted in order to deal with not only the three Big Data dimensions, but also with others such as the veracity, the variability and complexity. The study evidenced that more research efforts are needed for activities, such as data provenance, data quality and data security. In this regard, most of the research questions that have been proposed are related to the use of

data provenance to benefit other Big Data activities. Other research questions that arose were the preparation of data with complex structures, and the development of models and methodologies to design and implement Big Data Pipelines. The study also concluded that the most time-consuming activities in a Big Data Pipeline are data preparation and data acquisition. These are the first tasks that must be carried out before the data analysis activity, which is the cornerstone of the value extraction process.

Finally, a framework to facilitate the data preparation related to the transformation of complex data structure has been proposed in this master thesis. A versatile DSL has been designed in order to ease data preparation tasks. This proposal has been tested in a Big Data environment, and promising results have been obtained applying the solution to a real scenario.

5.2 Future Work

The study carried out in this work proves that more research efforts are required in the Big Data paradigm. Several research queries have been detected in this mater thesis. One of the lines of research that could have a great impact on the value extraction process is the *Data Quality, Provenance and security*. The design of models and techniques to facilitate the implementation of data provenance and its integration with other activities of the Big Data Pipeline, is a work that will be carried out in the future.

Another great advance would be the creation of methodologies to *Align Big Data Pipelines with Business Requirements*, facilitating the design and implementation of Big Data Pipelines that adapt the organization needs when they require the deployment of a Big Data system. This would contribute to the standardization of the Big Data process and costs reduction.

Regarding the proposal presented in Chapter 4 related to data integration, the framework that has been developed can be improved in many aspects. For example, the inclusion of conditional operators, the generalization to facilitate the integration with more Big Data platforms, the incorporation of mechanisms to assess the quality of the data, and the implementation of data provenance techniques to keep track of the transformations.

References

[1] A. Taherkordi, F. Eliassen, and G. Horn, "From IoT big data to IoT big services," in *Proceedings of the Symposium on Applied Computing - SAC '17*, (New York, New York, USA), pp. 485–491, ACM Press, 2017.

[2] A. Grillenberger and R. Romeike, "Big Data – Challenges for Computer Science Education," pp. 29–40, Springer, Cham, 2014.

[3] P. Sondergaard, "Introducing Infonomics - Peter Sondergaard," 2017. https://blogs.gartner.com/peter-sondergaard/introducing-infonomics/, Last accessed on 2019-01-30.

[4] M. Rüßmann, M. Lorenz, P. Gerbert, M. Waldner, J. Justus, P. Engel, and M. Harnisch, "Future of Productivity and Growth in Manufacturing," *Boston Consulting*, no. April, 2015. https://www.bcg.com/publications/2015/engineered_products_project_business_industry_4_future_productivity_growth_manufacturing_industries.aspx, Last accessed on 2018-12-19.

[5] J. Manyika, M. Chui, B. Brown, J. Bughin, R. Dobbs, C. Roxburgh, and A. H. Byers, "Big data: The next frontier for innovation, competition, and productivity | McKinsey," 2011. https://www.mckinsey.com/business-functions/digital-mckinsey/our-insights/big-data-the-next-frontier-for-innovation, Last accessed on 2018-12-21.

[6] E. Curry, "The big data value chain: Definitions, concepts, and theoretical approaches," in *New Horizons for a Data-Driven Economy: A Roadmap for Usage and Exploitation of Big Data in Europe*, pp. 29–37, Cham: Springer International Publishing, 2016.

[7] A. Oussous, F.-Z. Benjelloun, A. Ait Lahcen, and S. Belfkih, "Big Data technologies: A survey," *Journal of King Saud University - Computer and Information Sciences*, vol. 30, pp. 431–448, oct 2018.

[8] D. Laney, "3D data management: Controlling data volume, velocity, and variety," *META Group*, 2001.

[9] I. Lee, "Big data: Dimensions, evolution, impacts, and challenges," *Business Horizons*, vol. 60, pp. 293–303, may 2017.

[10] D. P. Acharjya and K. Ahmed, "A Survey on Big Data Analytics: Challenges, Open Research Issues and Tools," *International Journal of Advanced Computer Science and Applications*, vol. 7, no. 2, 2016.

[11] M. Obitko, V. Jirkovský, and J. Bezdíček, "Big data challenges in industrial automation," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 8062 LNAI, pp. 305–316, 2013.

[12] A. Gandomi and M. Haider, "Beyond the hype: Big data concepts, methods, and analytics," *International Journal of Information Management*, vol. 35, pp. 137–144, apr 2015.

[13] S. Nadal, V. Herrero, O. Romero, A. Abelló, X. Franch, S. Vansummeren, and D. Valerio, "A software reference architecture for semantic-aware Big Data systems," *Information and Software Technology*, vol. 90, pp. 75–92, oct 2017.

[14] K. Zhou, T. Liu, and L. Zhou, "Industry 4.0: Towards future industrial opportunities and challenges," in *2015 12th International Conference on Fuzzy Systems and Knowledge Discovery, FSKD 2015*, pp. 2147–2152, IEEE, aug 2016.

[15] I. Yaqoob, I. A. T. Hashem, A. Gani, S. Mokhtar, E. Ahmed, N. B. Anuar, and A. V. Vasilakos, "Big data: From beginning to future," *International Journal of Information Management*, vol. 36, no. 6, pp. 1231–1247, 2016.

[16] P. Ceravolo, A. Azzini, M. Angelini, T. Catarci, P. Cudré-Mauroux, E. Damiani, A. Mazak, M. Van Keulen, M. Jarrar, G. Santucci, K. U. Sattler, M. Scannapieco, M. Wimmer, R. Wrembel, and F. Zaraket, "Big Data Semantics," *Journal on Data Semantics*, vol. 7, pp. 65–85, jun 2018.

[17] C. A. Ardagna, V. Bellandi, P. Ceravolo, E. Damiani, M. Bezzi, and C. Hebert, "A Model-Driven Methodology for Big Data Analytics-as-a-Service," in *Proceedings - 2017 IEEE 6th International Congress on Big Data, BigData Congress 2017*, pp. 105–112, IEEE, jun 2017.

[18] P. Pääkkönen and D. Pakkala, "Reference Architecture and Classification of Technologies, Products and Services for Big Data Systems," *Big Data Research*, vol. 2, pp. 166–186, dec 2015.

[19] K. Lyko, M. Nitzschke, and A. C. Ngonga Ngomo, "Big Data Acquisition," in *New Horizons for a Data-Driven Economy: A Roadmap for Usage and Exploitation of Big Data in Europe*, pp. 39–61, Cham: Springer International Publishing, 2016.

[20] M. Shah, "Big Data and the Internet of Things," pp. 207–237, Springer, Cham, 2016.

[21] S. Poornima and M. Pushpalatha, "A journey from big data towards prescriptive analytics," *ARPN Journal of Engineering and Applied Sciences*, vol. 11, no. 19, pp. 11465–11474, 2016.

[22] G. Mansingh, K. M. Osei-Bryson, L. Rao, and M. McNaughton, "Data preparation: Art or science?," in *Proceedings of the 2016 International Conference on Data Science and Engineering, ICDSE 2016*, pp. 1–6, IEEE, aug 2017.

[23] N. Singh Gill, "Data Preparation, Preprocessing, Wrangling in Deep Learning - XenonStack," 2017. https://www.xenonstack.com/blog/data-science/preparation-wrangling-machine-learning-deep/, Last accessed on 2019-01-08.

[24] X. L. Dong and D. Srivastava, "Big Data Integration," *Synthesis Lectures on Data Management*, vol. 7, pp. 1–198, feb 2015.

[25] B. Arputhamary and L. Arockiam, "Data Integration in Big Data Environment," *Bonfring International Journal of Data Mining*, vol. 5, pp. 01–05, feb 2015.

[26] Y. Zheng, "Methodologies for Cross-Domain Data Fusion: An Overview," *IEEE Transactions on Big Data*, vol. 1, pp. 16–34, mar 2015.

[27] L. Guzenda, "Information Fusion vs. Data Integration," 2015. https://www.objectivity.com/information-fusion-and-data-integration-fast-vs-batch/, Last accessed on 2019-01-09.

[28] Y. F. Solahuddin and W. Ismail, "Data fusion for reducing power consumption in Arduino-Xbee wireless sensor network platform," in *2014 International Conference on Computer and Information Sciences (ICCOINS)*, pp. 1–6, IEEE, jun 2014.

[29] P. Cong and Z. Xiaoyi, "Research and Design of Interactive Data Transformation and Migration System for Heterogeneous Data Sources," in *2009 WASE International Conference on Information Engineering*, pp. 534–536, IEEE, jul 2009.

[30] S. H. A. El-Sappagh, A. M. A. Hendawi, and A. H. El Bastawissy, "A proposed model for data warehouse ETL processes," *Journal of King Saud University - Computer and Information Sciences*, vol. 23, pp. 91–104, jul 2011.

[31] S. Batasova, M. Efimova, I. Kholod, and A. Semenchenko, "Preparation of distributed heterogeneous data for data mining," in *2015 XVIII International Conference on Soft Computing and Measurements (SCM)*, pp. 205–207, IEEE, may 2015.

[32] P. Howard, "Data Preparation (self-service) – Bloor Research," 2018. https://www.bloorresearch.com/technology/data-preparation-self-service/, Last accessed on 2019-01-10.

[33] J. M. Hellerstein, J. Heer, and S. Kandel, "Self-Service Data Preparation: Research to Practice," *undefined*, 2018.

[34] Y. He, X. Chu, K. Ganjam, Y. Zheng, V. Narasayya, and S. Chaudhuri, "Transform-data-by-example (TDE): An Extensible Search Engine for Data Transformations," *Proceedings of the VLDB Endowment*, vol. 11, pp. 1165–1177, jun 2018.

[35] Z. Jin, M. R. Anderson, M. Cafarella, and H. V. Jagadish, "Foofah: Transforming Data By Example," in *Proceedings of the 2017 ACM International Conference on Management of Data - SIGMOD '17*, (New York, New York, USA), pp. 683–698, ACM Press, 2017.

[36] I. Taleb, H. T. E. Kassabi, M. A. Serhani, R. Dssouli, and C. Bouhaddioui, "Big Data Quality: A Quality Dimensions Evaluation," in *2016 Intl IEEE Conferences on Ubiquitous Intelligence & Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress (UIC/ATC/ScalCom/CBDCom/IoP/SmartWorld)*, pp. 759–765, IEEE, jul 2016.

[37] B. Schmarzo, *Big Data MBA - Driving Business Strategies with Data Science*, pp. 85–86. WIley, 2016.

[38] Chen, Chiang, and Storey, "Business Intelligence and Analytics: From Big Data to Big Impact," *MIS Quarterly*, vol. 36, no. 4, p. 1165, 2012.

[39] B. Schmarzo, *Big Data MBA - Driving Business Strategies with Data Science*, p. 88. WIley, 2016.

[40] B. Schmarzo, *Big Data MBA - Driving Business Strategies with Data Science*, pp. 91–93. WIley, 2016.

[41] B. Schmarzo, *Big Data MBA - Driving Business Strategies with Data Science*, pp. 89–90. WIley, 2016.

[42] V. Dhar and Vasant, "Data science and prediction," *Communications of the ACM*, vol. 56, pp. 64–73, dec 2013.

[43] B. Schmarzo, *Big Data MBA - Driving Business Strategies with Data Science*, p. 88. WIley, 2016.

[44] F. Sancho, "Introducción al Aprendizaje Automático - Fernando Sancho Caparrini." http://www.cs.us.es/{~}fsancho/?e=75, Last accessed on 2019-01-29.

[45] M. M. Najafabadi, F. Villanustre, T. M. Khoshgoftaar, N. Seliya, R. Wald, and E. Muharemagic, "Deep learning applications and challenges in big data analytics," *Journal of Big Data*, vol. 2, p. 1, dec 2015.

[46] L. Parody, Á. J. Varela-Vaca, M. T. Gómez-López, and R. M. Gasca, "FABIOLA: Defining the Components for Constraint Optimization Problems in Big Data environment," in *Information System Development - Improving Enterprise Communication, [Proceedings of the 26th International Conference on Information Systems Development, ISD 2017, Larnaca, Cyprus]*, 2017.

[47] T. Becker, "Big Data Usage," in *New Horizons for a Data-Driven Economy*, pp. 143–165, Cham: Springer International Publishing, 2016.

[48] M. Strohbach, J. Daubert, H. Ravkin, and M. Lischka, "Big Data Storage," in *New Horizons for a Data-Driven Economy*, pp. 119–141, Cham: Springer International Publishing, 2016.

[49] K. Shvachko, H. Kuang, S. Radia, and R. Chansler, "The Hadoop Distributed File System," in *2010 IEEE 26th Symposium on Mass Storage Systems and Technologies (MSST)*, pp. 1–10, IEEE, may 2010.

[50] C. Batini and M. Scannapieco, *Data and information quality: dimensions, principles and techniques*. 2016.

[51] International Organization for Standardization, "ISO/IEC 25012 - Data Quality model," tech. rep., 2008.

[52] A. Freitas and E. Curry, "Big Data Curation," in *New Horizons for a Data-Driven Economy*, pp. 87–118, Cham: Springer International Publishing, 2016.

[53] A. Siddiqa, I. A. T. Hashem, I. Yaqoob, M. Marjani, S. Shamshirband, A. Gani, and F. Nasaruddin, "A survey of big data management: Taxonomy and state-of-the-art," *Journal of Network and Computer Applications*, vol. 71, pp. 151–166, aug 2016.

[54] O. Kwon, N. Lee, and B. Shin, "Data quality management, data usage experience and acquisition intention of big data analytics," *International Journal of Information Management*, vol. 34, no. 3, pp. 387–394, 2014.

[55] A. Alkhalil and R. A. Ramadan, "IoT Data Provenance Implementation Challenges," *Procedia Computer Science*, vol. 109, pp. 1134–1139, jan 2017.

[56] C. A. Ardagna, P. Ceravolo, and E. Damiani, "Big data analytics as-a-service: Issues and challenges," in *Proceedings - 2016 IEEE International Conference on Big Data, Big Data 2016*, pp. 3638–3644, IEEE, dec 2016.

[57] "Big Data: A New World of Opportunities," 2012.

[58] J. C. Duarte, M. C. R. Cavalcanti, I. de Souza Costa, and D. Esteves, "An interoperable service for the provenance of machine learning experiments," in *Proceedings of the International Conference on Web Intelligence - WI '17*, (New York, New York, USA), pp. 132–138, ACM Press, 2017.

[59] J. Wang, D. Crawl, S. Purawat, M. Nguyen, and I. Altintas, "Big data provenance: Challenges, state of the art and opportunities," in *2015 IEEE International Conference on Big Data (Big Data)*, vol. 2015, pp. 2509–2516, IEEE, oct 2015.

[60] I. S. of Information Sciences, "Foundations of Data Curation," 2018. https://ischool.illinois.edu/degrees-programs/courses/is531, Last accessed on 2018-12-19.

[61] M. Stonebraker, M. Stonebraker, G. Beskales, A. Pagan, D. Bruckner, M. Cherniack, S. Xu, V. Analytics, I. F. Ilyas, and S. Zdonik, "Data Curation at Scale: The Data Tamer System," *IN CIDR 2013*, 2013.

[62] C. Rusbridge, P. Buneman, P. Burnhill, D. Giaretta, S. Ross, L. Lyon, and M. Atkinson, "The Digital Curation Centre: A Vision for Digital Curation," in *2005 IEEE International Symposium on Mass Storage Systems and Technology*, pp. 31–41, IEEE.

[63] S. Choi, J. Seo, M. Kim, S. Kang, and S. Han, "Chrological big data curation: A study on the enhanced information retrieval system," *IEEE Access*, vol. 5, pp. 11269–11277, 2017.

[64] Y. Demchenko, C. Ngo, C. De Laat, P. Membrey, and D. Gordijenko, "Big security for big data: Addressing security challenges for the big data infrastructure," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 8425 LNCS, pp. 76–94, Springer, Cham, 2014.

[65] D. S. Terzi, R. Terzi, and S. Sagiroglu, "A survey on security and privacy issues in big data," in *2015 10th International Conference for Internet Technology and Secured Transactions (ICITST)*, pp. 202–207, IEEE, dec 2015.

[66] Y. Demchenko, C. de Laat, and P. Membrey, "Defining architecture components of the Big Data Ecosystem," in *2014 International Conference on Collaboration Technologies and Systems (CTS)*, pp. 104–112, IEEE, may 2014.

[67] Microsoft, "Big data architectures | Microsoft Docs," 2018. https://docs.microsoft.com/en-us/azure/architecture/data-guide/big-data/, Last accessed on 2019-01-16.

[68] D. Singh and C. K. Reddy, "A survey on platforms for big data analytics," *Journal of Big Data*, vol. 2, p. 8, dec 2015.

[69] N. Marz and J. J. O. Warren, *Big data: principles and best practices of scalable real-time data systems*. 2015.

[70] J. Dean and S. Ghemawat, "MapReduce: Simplified Data Processing on Large Clusters," 2004.

[71] J. Kreps, "Questioning the Lambda Architecture - O'Reilly Media," 2014. https://www.oreilly.com/ideas/questioning-the-lambda-architecture, Last accessed on 2018-12-16.

[72] V. Persico, A. Pescapé, A. Picariello, and G. Sperlí, "Benchmarking big data architectures for social networks data processing using public cloud platforms," *Future Generation Computer Systems*, vol. 89, pp. 98–109, dec 2018.

[73] M. Marjani, F. Nasaruddin, A. Gani, A. Karim, I. A. T. Hashem, A. Siddiqa, and I. Yaqoob, "Big IoT Data Analytics: Architecture, Opportunities, and Open Research Challenges," *IEEE Access*, vol. 5, pp. 5247–5261, 2017.

[74] C. Cecchinel, M. Jimenez, S. Mosser, and M. Riveill, "An Architecture to Support the Collection of Big Data in the Internet of Things," in *2014 IEEE World Congress on Services*, pp. 442–449, IEEE, jun 2014.

[75] M. Villari, A. Celesti, M. Fazio, and A. Puliafito, "AllJoyn Lambda: An architecture for the management of smart environments in IoT," in *2014 International Conference on Smart Computing Workshops*, pp. 9–14, IEEE, nov 2014.

[76] M. Kiran, P. Murphy, I. Monga, J. Dugan, and S. S. Baveja, "Lambda architecture for cost-effective batch and speed big data processing," in *2015 IEEE International Conference on Big Data (Big Data)*, pp. 2785–2792, IEEE, oct 2015.

[77] M. Strohbach, H. Ziekow, V. Gazis, and N. Akiva, "Towards a Big Data Analytics Framework for IoT and Smart City Applications," pp. 257–282, Springer, Cham, 2015.

[78] O.-C. Marcu, A. Costan, G. Antoniu, M. S. Perez-Hernandez, R. Tudoran, S. Bortoli, and B. Nicolae, "Towards a unified storage and ingestion architecture for stream processing," in *2017 IEEE International Conference on Big Data (Big Data)*, pp. 2402–2407, IEEE, dec 2017.

[79] J. Stefanowski, K. Krawiec, and R. Wrembel, "Exploring complex and big data," *International Journal of Applied Mathematics and Computer Science*, vol. 27, pp. 669–679, dec 2017.

[80] B. Wanders, M. van Keulen, and P. van der Vet, "Uncertain Groupings: Probabilistic Combination of Grouping Data," pp. 236–250, Springer Publishers, sep 2015.

[81] I. Taleb and M. A. Serhani, "Big Data Pre-Processing: Closing the Data Quality Enforcement Loop," in *2017 IEEE International Congress on Big Data (BigData Congress)*, pp. 498–501, IEEE, jun 2017.

[82] I. Suriarachchi and B. Plale, "Provenance as Essential Infrastructure for Data Lakes," pp. 178–182, Springer, Cham, 2016.

[83] L. Parody, Á. J. Varela-Vaca, M. T. G. López, and R. M. Gasca, "FABIOLA: defining the components for constraint optimization problems in big data environment," in *Information Systems Development: Advances in Methods, Tools and Management - Proceedings of the 26th International Conference on Information Systems Development, ISD 2017, Larnaca, Cyprus, University of Central Lancashire Cyprus, September 6-8, 2017*, 2017.

[84] D. Riehle, Dirk, Riehle, and Dirk, "Composite design patterns," *ACM SIGPLAN Notices*, vol. 32, pp. 218–228, oct 1997.

[85] E. D. Reilly, A. Ralston, and D. Hemmendinger, "Backus-Naur form (BNF)," in *Encyclopedia of Computer Science*, pp. 129–131, Wiley, 2003.

[86] Mesosphere, "Dc/os," 2019. https://mesosphere.com/about/, Last accessed on 2019-01-17.

[87] Apache Foundation, "Apache mesos," 2019. http://mesos.apache.org/, Last accessed on 2019-01-17.

[88] Apache Foundation, "Apache Spark 2.3.1," 2019. https://spark.apache.org/, Last accessed on 2019-01-17.

[89] M. Zaharia, R. S. Xin, P. Wendell, T. Das, M. Armbrust, A. Dave, X. Meng, J. Rosen, S. Venkataraman, M. J. Franklin, A. Ghodsi, J. Gonzalez, S. Shenker, and I. Stoica, "Apache spark: a unified engine for big data processing," *Commun. ACM*, vol. 59, no. 11, pp. 56–65, 2016.

[90] Apache Foundation, “Hadoop,” 2019. https://hadoop.apache.org/, Last accessed on 2019-01-17.

[91] Mongo, “Mongodb,” 2019. https://www.mongodb.com/, Last accessed on 2019-01-17.