Chapter 8
DIPAR: A Framework for Implementing Big Data Science in Organizations

Luis Eduardo Bautista Villalpando, Alain April and Alain Abran

Abstract Cloud computing (CC) is a technology aimed at processing and storing very large amounts of data, which are also referred to as big data (BD). Although this is not the only aim of the cloud paradigm, one of the most important challenges in CC is how to process and deal with the BD. By the end of 2012, the amount of data generated was approximately 2.8 zettabytes (ZB), i.e., 2.8 trillion GB. One of the areas that contribute to the analysis of BD is referred to as *data science*. This new study area, also called big data science (BDS), has recently become an important topic in organizations because of the value it can generate, both for themselves and for their customers. One of the challenges in implementing BDS is the current lack of information to help in understanding this new study area. In this context, this chapter presents the define-ingest-preprocess-analyze-report (DIPAR) framework, which proposes a means to implement BDS in organizations and defines its requirements and elements. The framework consists of five stages define, ingest, preprocess, analyze, and report. It is based on the ISO 15939 Systems and Software Engineering—Measurement process standard, the purpose of which is to collect, analyze, and report data relating to the products to be developed.

Keywords Big data science · Data cleaning · DIPAR framework · ISO 15939 · Security · System requirements

L. E. B. Villalpando (✉)
Department of Electronic Systems, Autonomous University of Aguascalientes,
Av. Universidad 940, Ciudad Universitaria, Aguascalientes, AGS, Mexico
e-mail: lebautis@correo.uaa.mx

A. April · A. Abran · L. E. B. Villalpando
Department of Software Engineering and Information Technology,
ETS—University of Quebec, 1100 Notre-Dame St., Montreal, Canada
e-mail: alain.april@etsmtl.ca

A. Abran
e-mail: alain.abran@etsmtl.ca

Z. Mahmood (ed.), *Continued Rise of the Cloud,* Computer Communications and Networks, DOI 10.1007/978-1-4471-6452-4_8,

8.1 Introduction

Cloud computing (CC) is a technology aimed at processing and storing very large amounts of data. According to the ISO subcommittee 38, the CC study group, CC is a paradigm for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable cloud resources accessed through services which can be rapidly provisioned and released with minimal management effort or service provider interaction [1].

One of the most important challenges in CC is how to process large amounts of data (also known as big data—BD) in an efficient and reliable way. In December 2012, the International Data Corporation (IDC) stated that, by the end of 2012, the total data generated was 2.8 zettabytes (ZB), i.e., 2.8 trillion GB [2]. Furthermore, the IDC predicts that the total data generated by 2020 will be 40 ZB. This is roughly equivalent to 5.2 terabytes (TB) of data generated by every human being alive in that year. In addition, according to the report, only 0.5 % of the data have been analyzed up to the present time, and one-quarter of all the currently available data may contain valuable information. This means that *BD processing* will be a highly relevant topic in the coming years.

One of the main areas contributing to the analysis of BD is *data science* (DS). Although this term has emerged only recently, it has a long history, as it is based on techniques and theories from fields, such as mathematics, statistics, data engineering, etc. [3]. The integration of these fields into the BD paradigm has resulted in a new study area called *big data science* (BDS).

BDS has recently become a very important topic in organizations because of the value it can generate, both for themselves and for their customers. One of the main challenges in BDS is the current lack of information to help in understanding, structuring, and defining how to integrate this study area into organizations and how to develop processes for its implementation. BDS involves implementation challenges not faced in DS, such as the integration of large amounts of data from different sources, data transformation, storage, security, the analysis of large data sets using high-performance processing technologies, and the representation of analysis results (visualization), to mention only a few.

This chapter presents the define-ingest-preprocess-analyze-report (DIPAR) framework, which proposes a means to implement BDS in organizations, and defines the requirements and elements involved. The framework consists of five stages: *Define*, *Ingest*, *Preprocess*, *Analyze*, *and Report (DIPAR)*, which we describe here. We also explain how to implement these stages, along with the components of the framework.

The rest of this chapter is organized as follows. Section 8.2 presents an overview of BDS, including its definition and history, as well as its relationships with other study areas, like data mining (DM) and data analysis (DA). Section 8.3 presents the ISO 15939 Systems and Software Engineering—Measurement process standard, the purpose of which is to collect, analyze, and report data relating to products to be developed. Section 8.4 constitutes the core of this chapter, in which we present our proposal of the DIPAR framework as a means to implement BDS in organizations.

Section 8.5 describes the relationships between this framework and the measurement processes defined in the ISO 15939 measurement process standard. Section 8.6 presents a case study in which the DIPAR framework is used to develop a BD product for the performance analysis of CC applications. Finally, Sect. 8.7 presents a summary and the conclusions of this chapter.

8.2 Big Data Science

The term big data science has been in common usage for only about 3 years, but it has evolved, in part, from the term *data analysis*. In 1962, Tukey [4] writes that, with the evolution of mathematical statistics, it will be possible to apply them to "very extensive data," which is the central interest in DA. Moreover, he points out that DA includes, among other things: procedures for analyzing data, techniques for interpreting the results of those procedures, ways of planning the data gathering process to make analysis of the data easier, and so on.

In recent years, DM has been the area of knowledge that has been responsible for DA in organizations. Authors like Han et al. [5] describe DM as an interdisciplinary subject which includes an iterative sequence of steps for what he calls *knowledge discovery*. These steps are data cleaning, data integration, data selection, data transformation, pattern evaluation, and the presentation of results. Han explains that these steps can be summarized in the extraction/transformation/loading (ETL) process. Extraction is the stage in which data are collected from outside sources, transformation is the stage in which methods and functions are applied to data in order to generate valuable information, and loading is the stage in which the data are inputted into the end target to generate output reports.

Although the ETL process has been applied in organizations for some years, this approach cannot be used in its entirety in BD, because the traditional data warehouse tools and processes are not designed to work on very large amounts of data. Some authors, like Lin [6], contend that "big data mining" is about much more than what most academics would consider simply DM. He goes on to say that a significant amount of tooling and infrastructure is required to operationalize vague strategic directives into concrete, solvable problems with clearly defined indicators of success. Other authors, like Thusoo et al. [7], note that the BD processing infrastructure has to be flexible enough to support optimal algorithms and techniques for the very different query workloads. Moreover, Thusoo emphasizes that what makes this task more challenging is that the data under consideration continue to grow rapidly. In one example, in 2012 alone, Facebook generated more than 500 TB of new data every day.

8.3 Big Data Science as a Measurement Process

One of the most important challenges for organizations is to turn available data into final products which generate value and create a competitive advantage for enterprises and institutions. Meeting this challenge is vital to the development of measurement

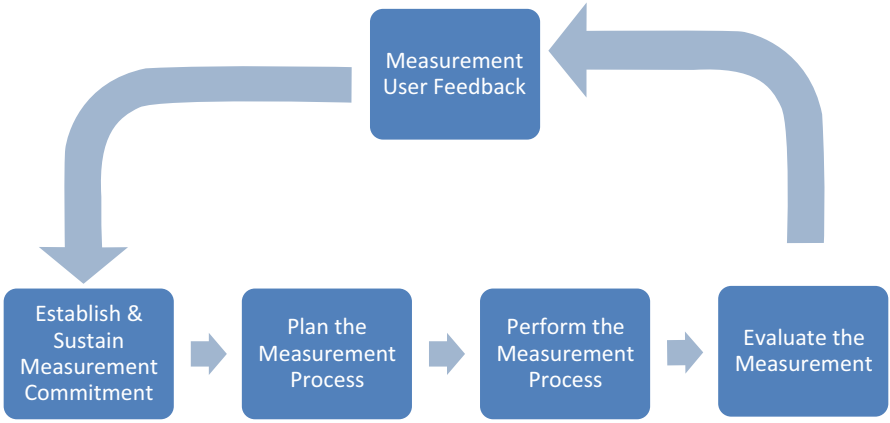


Fig. 8.1 Sequence of activities in a measurement process (Adapted from the ISO 15939 measurement process model [8])

processes that support the analysis of information related to the original data, with a view to defining the types of products that can be developed. According to ISO 15939 [8] *Systems and Software Engineering—Measurement process*, the purpose of a measurement process is to collect, analyze, and report data relating to the products developed and the processes implemented within the organizational unit to support effective management of the measurement process and to objectively demonstrate the quality of the products.

ISO 15939 defines a sequence of four activities to develop such measurements, which include establish and sustain measurement commitment, plan the measurement process, perform the measurement process, and evaluate the measurement. These activities are performed in an iterative cycle that allows for continuous feedback and improvement of the measurement process, as shown in Fig. 8.1.

The activities performed during the measurement process are described below:

Establish and sustain measurement commitment. This activity consists of two tasks: (1) define the requirements for measurement (2) assign resources. Defining the requirements for measurement involves defining the scope of measurement as a single project, a functional area, the whole enterprise, etc., as well as the commitment of management and staff to the measurement process. This means that the organizational unit should demonstrate its commitment through policies, allocation of responsibilities, budget, training, etc. Assigning resources involves the allocation of responsibilities to individuals, as well as the provision of resources to plan the measurement process.

- Plan the measurement process. This activity consists of a series of tasks, such as identifying information needs, selecting measures, defining data collection, and defining the criteria for evaluating product and process information. It also includes the activities related to reviewing, approving, and providing resources for measurement tasks.

Fig. 8.2 Stages to develop during implementation of the DIPAR framework

- Perform the measurement process. This activity consists of the tasks defined in the planning activity, along with the following sub activities: integrate procedures, collect data, analyze data, develop information products, and communicate results.
- Evaluate the measurement. This activity consists of evaluating information products against the specified evaluation criteria and of determining the strengths and weaknesses of the information products and the measurement process. This activity must also identify potential improvements to the information products. For instance, changing the format of an indicator, changing from a linear measure to an area measure, minutes to hours, or a line of code to a size measure, etc.

The next section presents the DIPAR framework, its stages, and its implementation process. It also describes the relationships that exist between the stages and how the framework is integrated into the measurement processes defined in ISO 15939.

8.4 The DIPAR Framework

The DIPAR framework integrates the four activities described in ISO 15939, and its main objective is to design BD products that have a high impact on organizational performance. Figure 8.2 depicts the five stages to be executed during the implementation of the DIPAR framework, as well as the order in which should be executed.

The following subsections describe each stage of the DIPAR framework and the elements they involve.

8.4.1 Define the New Product to Develop

The first step in the DIPAR framework is to define whether or not a new BD product is necessary. If it is not, all the analytical work developed to create the product will be a waste of time and resources. Clearly, it is sometimes not possible to establish the type of product to be developed, because there is no knowledge available on the type of data that can be collected and analyzed. Patil [9] suggests that this issue can be resolved by taking shortcuts in order to get products off the ground. He maintains that these shortcuts will enable products to survive to the finished state because they support good ideas that might not have seen the light of day otherwise, and that taking them will result in the development of more complex analytical techniques that will form a baseline for building better products in the future. Moreover, these basic ideas should be aligned with the strategic objectives of the organization, e.g., "*it is necessary to improve the user experience in the online store, in order to increase sales*." This strategy would enable these ideas to form the basis for building new products, such as recommender systems and prediction systems.

8.4.2 Ingest the Big Data System

In order to clearly define the boundaries of the new product, large amounts of data need to be analyzed. One of the main challenges of ingesting a BD system is to define the ingestion sources, because most of the time data come from a number of sources, such as Web logs, databases, and different types of applications. This makes very difficult to know what type of data will be ingested by the BD system. For instance, an organization can gather behavioral data from users from very different sources, like Web page logs users visit, links users click, social media, the location systems included in mobile devices, etc. In addition, many of these data sources (services) are loosely coordinated systems which lead to the creation of a large number of isolated data stores. This distributed scheme makes it difficult to know the type of data that is being collected, as well as its state. In addition, the services provided by different systems change over time and as functionalities evolve; sometimes systems are merged into newer ones or replaced entirely. All these issues result in inaccuracies in the data to be analyzed, which must be kept in mind in the BD ingestion process.

One solution to this problem is to use BD software that is designed specifically to collect and aggregate data from different sources. Projects like Flume [10] and Scribe [11] allow large amounts of log data to be collected, aggregated, and moved from many different sources to a centralized data store. Moreover, since data sources are customizable, this type of software can be used to transport massive quantities of event data, including, but not limited to, network traffic data, data generated by social media, and email messages, in fact, pretty much any data source imaginable.

8.4.3 Big Data Preprocessing

One of the main problems that arises following the ingestion of a BD system is the *cleanliness of data*. This problem calls for the quality of the data to be verified prior to performing *BD Analysis (BDA)*. According to Lin [6], during the data collection process for Twitter, all the large, real-world data sets required data cleaning to render them usable. Among the most important data quality issues to consider during data cleaning in a BDS are corrupted records, inaccurate content, missing values, and formatting inconsistencies, to name a few. Another important issue in data quality assurance in BD preprocessing is formatting inconsistencies, caused by the very different forms that data can take. For example, the property *product ID* could be labeled *product_id* in one data service and *productID* in other service. Furthermore, the data type for this property could be assigned a numeric value in the first case and an alphanumeric value in the second case.

Consequently, one of the main challenges at the preprocessing stage is how to structure data in standard formats so that they can be analyzed more efficiently. This is often easier said than done: during the process of structuring and merging data into common formats, there is a risk of losing valuable information. This challenge is a current topic of investigation among researchers.

Another issue to address before embarking on BDA is to determine what data fields are the most relevant, in order to construct analysis models [12]. One way to resolve this issue is to sample the data to obtain an overview of the type of data collected, in order to understand the relationships among features spread across multiples sources. Of course, training models on only a small fraction of data does not always give an accurate indication of the model's effectiveness at scaling up [6].

8.4.4 Big Data Analysis

Once the data have been preprocessed, they are analyzed to obtain relevant results. For this, it is necessary to develop models which can be used in the creation of new products. One of the main problems arising during the design of such models is to recognize which of the available data are the most relevant to an analysis task. During a study of the BDA implementation process in organizations, Kandel et al. [12] found that almost 60 % of data scientists have difficulty understanding the relationships between features spread across multiple databases. He also found that the main challenge in this process is feature selection, which is an important step in the development of accurate models. Sampling is a good way to address these challenges.

In addition, according to Kandel, most data scientists have a problem with the size of their data sets. This is because the majority of the existing analysis packages, tools, and algorithms do not scale-up with BD sets. One way to solve this problem is to use one of the new BD technologies, which make it possible to process and analyze large data sets in a reasonable amount of time. For example, with Hive [13], this type of task can be performed very rapidly. Hive is a data warehousing framework created

at Facebook for reporting ad hoc queries and analyzing their repositories. Other products, like Mahout [14], help to build scalable machine learning libraries which can be used on large data sets. Mahout supports four use cases in which machine learning techniques are used: recommendation mining, clustering, classification, and market basket analysis.

Once it becomes feasible to develop complex models and algorithms for DA, it is possible to create products with added value for the organization. However, to establish the direction to be taken during the product development process, we need to understand the results of the previous analyses. For example, once Amazon had analyzed its large data set, they found that they could use the historical record of Web pages visited by users to create a recommender system, for example: "*People who viewed product X also viewed product Y.*"

It is clear that a mechanism is required for presenting the analysis results so that they can be studied and understood, and, also communicated to the stakeholders involved in the design of the product. In the next section, we describe aspects that must be considered when reporting the analysis results.

8.4.5 Reporting of Results (Visualization)

Once BD are ingested, preprocessed, and analyzed, users need to be able to access and evaluate the results, which must be presented in such a way that they are readily understood. Often they are presented in statistical charts and graphs that contain too much information which is not descriptive enough for the end user. Although a number of BD analysts still deliver their results only in static reports, some end users complain that this system is inflexible and does not allow for interactive verification in real time [12]. According to the Networked European Software and Services Initiative (NESSI), in its technical paper entitled, "*Big Data, A New World of Opportunities*" [15], reports generated from DA can be thought of as documents. These documents frequently contain varying forms of media in addition to a textual representation. They add that the interface through which this complex information is communicated needs to be responsive to human needs ("humane"), user-friendly, and closely linked to the knowledge of the users. To achieve this, the NESSI proposes the use of Visual Analytics (VA), which combines the strengths of human and electronic data processing. The main objective of VA is to develop knowledge, methods, technologies, and practices that exploit human capabilities and the capacities of electronic data processing. They list the key features of VA, which are:

- Emphasizes DA, problem solving, and/or decision making
- Leverages computational processing by applying automated techniques for data processing, knowledge discovery algorithms, etc.
- Encourages the active involvement of a human in the analytical process through interactive visual interfaces
- Supports the provenance of analytical results
- Supports the communication of analytical results to the appropriate recipients

Furthermore, authors like Yau [16] maintain that data visualization is like a story in which the main character is a user who can take two paths. A story of charts and graphs might read much like a textbook; however, a story with context, relationships, interactions, patterns, and explanations reads more like a novel. Nevertheless, the former is not necessarily better than the latter. What this author suggests is that the content should be presented in a format somewhere between a textbook and a novel, so that BD can be visualized, that is, facts are provided, but context as well.

At the same time, authors like Agrin [17] focus on the real challenges that BD visualization developers face and what should be avoided when implement BD visualization. Agrin maintains that simplicity must be the goal in data visualization, suggesting, at the risk of sounding regressive, that there are good reasons to work with charts that have been in continuous use since the eighteenth century. In addition, he notes that the bar chart is one of the best tools available for facilitating visual comparison, as it takes advantage of our innate ability to compare side-by-side lengths. Agrin [17] also lists a number of tools and strategies which can be useful in the design of data visualization methods:

- Do not dismiss traditional visualization choices, if they represent the best option for your data.
- Start with bar and line charts, and look further only when the data requires it.
- Have a good rationale for choosing other options.
- Compared to bar charts, bubble charts support more data points with a wider range of values; pies and doughnuts clearly indicate part-to-whole relationships; tree maps support categories organized hierarchically.
- Bar charts have the added bonus of being one of the easiest visualizations to create: an effective bar chart can be hand coded in HTML using nothing but the cascading style sheet (CSS) and minimal JavaScript, or one can be created in Excel with a single function.

To summarize, it is important to consider the type of results to be presented in determining what scheme of visual representation will be used. On the one hand, if we need to show the degree of relationships between persons, graph representation may be the best option. On the other hand, if we need to show the degree of influence of certain factors on the performance of CC systems, perhaps the best option is the bar chart.

In the next section, we present the relationships among the elements of the DIPAR framework and the ISO 15939 Systems and Software Engineering—Measurement process standard.

8.5 DIPAR Framework and ISO 15939 Measurement Process

One of the main characteristics of the DIPAR framework is that it was designed taking into account the ISO 15939 measurement process activities. Each stage presented in the DIPAR framework is mapped to the activities described in the ISO 15939

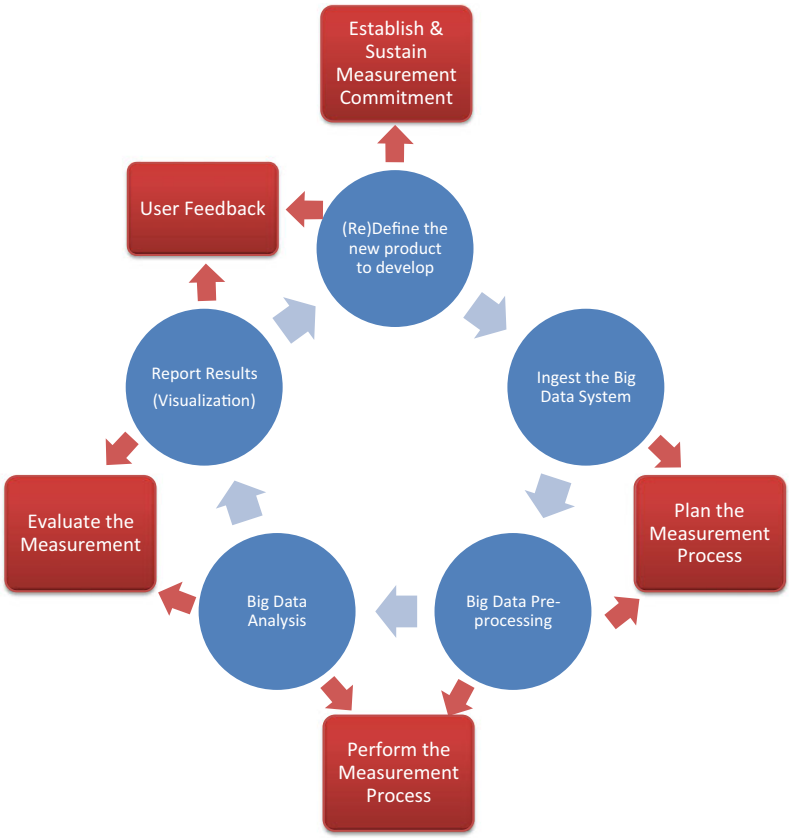


Fig. 8.3 Relationship between the DIPAR framework and the ISO 15939 standard

standard; both the stages and the activities follow the sequence defined in ISO 15939. Figure 8.3 shows the relationships that exist between the DIPAR framework stages and the activities defined in ISO 15939.

Once the DIPAR framework has been mapped to ISO 15939, it is necessary to present in a detailed form which of the stages described in the DIPAR framework are part of the ISO 15939 activities. Table 8.1 presents the relationships between the DIPAR framework and the ISO 15939 measurement process.

In the next section, we present a case study which uses the DIPAR framework to develop a BD product for analyzing the performance of CC applications.

Table 8.1 Relationship between the DIPAR framework and the ISO 15939 measurement process

ISO 15939 Activity	DIPAR Stage	Activities to Perform in the DIPAR Stages
1. Establish and sustain measurement commitment	1. Define the new BD product to develop	Define the new BD product requirements Align the product with the strategic objectives of the organization Define the scope of the product Devise a development plan Assign resources to the development of the product
2. Plan the measurement process	2. Ingest the big data system 3. Big data preprocessing	Define the data collection sources Sketch the type of data to collect Define the interfaces for data collection from the various data sources Verify data quality Perform data cleaning
3. Perform the measurement process	3. Big data preprocessing (cont.) 4. Big data analysis	Obtain an overview of the relationships among the collected data (e.g., sampling) Develop models and algorithms for the data analysis Implement the selected models using big data processing technologies
4. Evaluate the measurement	4. BDA (cont.) 5. Report the results (visualization)	Prepare the results in order to report them to the users Select the type of format to use to present results (graphs, charts, bar charts, etc.) Design flexible reports, in order to be able to update them in real time Design user-friendly interfaces to present results Support the results using a human-oriented analytical process
5. User feedback	6. Redefine the product to develop	Use the results to create or define more complex products, such as recommender systems, prediction systems, and market basket products, etc. Restructure new data to develop the new products

8.6 Case Study

8.6.1 Introduction

One of the most important challenges in delivering cloud services (CS) is to ensure that they are fault tolerant, as failures and anomalies can degrade these services and impact their quality, and even their availability. According to Coulouris et al. [18], a failure occurs in a distributed system (DS), like a CC system, when a process or communication channel departs from what is considered to be its normal or desired behavior. An anomaly is different, in that it slows down a part of a CC system without making it fail completely. It impacts the performance of tasks within nodes, and, consequently, of the system itself.

Developing products for CC systems, and more specifically for CC Applications (CCA), which propose a means to identify and quantify "normal application behavior," can serve as a baseline for detecting and predicting possible anomalies in the software (i.e., jobs in a cloud environment) that may impact cloud application performance.

The CS use different technologies for the data storage, processing, and development services they offer, through various frameworks for managing CCA. Hadoop is one of the technologies used most often in CS, because it offers open source tools and utilities for CC environments. Hadoop includes a set of libraries and subsystems which permit the storage of large amounts of information, enabling the creation of very large data tables or summarizing data with tools that are part of the data warehouse infrastructure. Although there are several kinds of application development framework for CC, such as GridGain, Hazelcast, and DAC, Hadoop has been widely adopted because of its open source implementation of the MapReduce programming model, which is based on Google's MapReduce framework [19].

According to Dean [19], programs written in MapReduce are automatically parallelized and executed on a large cluster of commodity machines. In addition, according to Lin's [20] approach to tackling large data problems today is to divide and conquer, which means that a large problem is broken down into smaller sub problems. Those sub problems can be tackled in parallel by different workers. For example, threads in a processor core, cores in a multi-core processor, multiple processors in a machine, or many machines in a cluster. Intermediate results from each individual worker are then combined to yield the final output.

CC systems in which MapReduce applications are executed are exposed to common-cause failures, which are a direct result of a common cause or a shared root cause, such as extreme environmental conditions, or operational or maintenance errors [21]. Some examples of common-cause failures in CC systems are memory failures, storage failures, and process failures. For this reason, it is necessary to develop a product capable of identifying and quantifying "normal application behavior" by collecting base measures specific to CCA performance, such as application processing times, the memory used by applications, the number of errors in a network transmission, etc.

In the next subsection, we present the implementation process of the DIPAR framework for the creation of a product that can identify and quantify the normal application behavior of CCA.

8.6.2 Define the Product to Develop

The first stage in the DIPAR framework implementation process is to define the product to be developed. Table 8.2 shows the BD product definition stage and the items involved in it .

Table 8.2 Product definition stage and the items involved

Product name: CCA performance analysis application	DIPAR stage: BD product definition
Item	Values
1. Product requirements	The product must improve CCA performance The product must include a performance measurement process (PMP) The PMP must be able to measure Hadoop performance characteristics
2. Product alignment with the strategic objectives of the organization	The product must improve the performance of the organization by increasing the quality of provision of services in BD processing
3. Scope of the product	The product must provide performance analysis for users, developers, and maintainers The product must be able to measure MapReduce and Hadoop system performance characteristics The product must not include analysis of elastic or virtualized cloud systems
4. Development plan definition	The product will be developed through the following steps: Install a Hadoop test cluster Collect system and application performance measures Develop a performance analysis model Report analysis model results
5. Resource allocation	Hadoop test cluster BD scientist MapReduce developer BD visualization developer

Table 8.3 BDS ingestion stage and the items involved

Product name: CCA performance analysis application	DIPAR stage: BD ingestion
Item	Values
Data types to be collected	Two data types must be collected: (a) Hadoop cluster measures (b) MapReduce application execution measures
Data source identification	Hadoop system logs MapReduce logs System monitoring tool measures (e.g., Ganglia, Nagios, etc.) MapReduce execution statistics
Interfaces for merging data	The data collected from the sources will be merged and stored in a BD repository like HBase [22]

8.6.3 Ingest the Big Data System

The second stage in the DIPAR framework implementation process is to ingest the BDS. In this stage, the type of data to collect is defined as well as their sources. Table 8.3 presents the elements involved in BDS ingestion.

Table 8.4 BDS preprocessing stage and the items involved

Product name: CCA performance analysis application	DIPAR stage: BD preprocessing
Item	Values
1. Data quality	The data collected from the various sources, such as logs, monitoring tools, and application statistics, are parsed and examined using the cleaning process provided by the Hadoop Chukwa [23] libraries Chukwa is a large-scale log collection and analysis system supported by the Apache Software Foundation
2. Data cleaning	The data cleaning process is performed using the Chukwa raw log collection and aggregation work flow In Chukwa, a pair of MapReduce jobs runs every few minutes, taking all the available logs files as input to perform the data cleaning process [24] The first job simply archives all the collected data, without processing it or interpreting it The second job parses out structured data from some of the logs and then cleans and loads those data into a data store (HBase)

8.6.4 Big Data Preprocessing

As already mentioned, one of the main problems that arises following the BDS ingestion stage is ensuring that the data are clean. To achieve this, preprocessing is necessary, in order to verify the quality of the data to be subjected to BDA. Table 8.4 presents the elements involved in preprocessing and the steps to be followed.

8.6.5 Big Data Analysis

Once the data have been preprocessed, they can be analyzed to obtain relevant results. In this case study, a performance measurement framework for CCA [25] is used, in order to determine the form in which the system performance characteristics should be measured. Table 8.5 presents the elements involved in the BDA stage and the steps required to execute BDA.

8.6.6 Reporting the Results (Visualization)

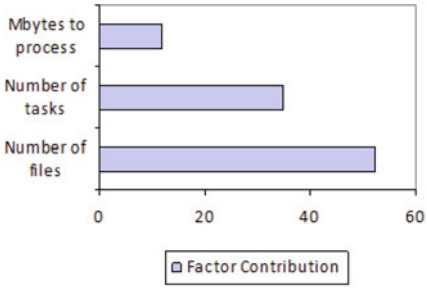
Once the BDS has been analyzed, the results are evaluated. They have to be presented in such a way that they are understood in statistical charts and graphs containing information that is descriptive for the end user. Table 8.6 presents the elements involved in the results reporting stage and the elements involved.

Table 8.5 BDA stage and the items involved

Product name: CCA performance analysis application	DIPAR stage: BD analysis
Item	Values
Overview of the relationships between collected data (sampling)	The Performance Measurement Framework for Cloud Computing [25] defines the elements necessary to measure the behavior of cloud systems using software quality concepts The framework determines that the performance efficiency and reliability concepts are closely related in performance measurement The framework determines five function categories for collecting performance measures, which are failure function, fault function, task application function, time function, and transmission function
Data analysis models and algorithms	In order to analyze and determine the types of relationships that exist in the measures collected from Hadoop, a methodology for the performance analysis of CCA is used [26] This methodology uses the Taguchi method for the design of experiments to identify the relationships between the various parameters (base measures) that affect the quality of CCA performance One of the goals of this framework is to determine what types of relationships exist between the various base measures. For example, what is the extent of the relationship between the CPU processing time and the amount of information to process?
Model implementation using BD processing technologies	Once the analysis method is determined, it is implemented by Apache's Pig and Hive technologies in order to apply it to the BD repository Apache Hive [13] is a data warehouse system for Hadoop that facilitates easy data summarization, ad hoc queries, and the analysis of large data sets stored in Hadoop-compatible file systems Apache Pig [27] is a platform for analyzing large data sets that consists of a high-level language for expressing data analysis programs, coupled with the infrastructure for evaluating these programs. The salient property of Pig programs is that their structure is amenable to substantial parallelization, which in turns enables them to handle very large data sets

Once the DIPAR framework implementation process was completed, the design team observed that the original product could be redefined to create a new product. This decision was based on the type of data and the results collected. Specifically, the results show a strong relationship between certain factors (base measures) and the performance of applications running on the cluster. Based on these results, we were able to develop a new performance analysis application. The new product has two main functions, the first as a recommender system, and the second as a fault

Table 8.6 Results reporting stage and the items involved

Product Name: *CCA Performance Analysis Application*	**DIPAR Stage:** *BD Report the Results*
Item	**Values**
1. Format to use to present the results	• Bar charts, line charts, and scatter charts were chosen to present the relationships between the various performance factors. • e.g. chart of factor contribution percentages 
2. User interfaces to present the results	• A Web-based data visualization scheme was selected to present the results. The JavaScript and D3.js libraries were selected to design the data visualization Web site.
3. A human analytical process to support the results	• The charts were supported with textual explanations and classroom presentations by a data scientist.

prediction system. The performance analysis results can also be used to implement different machine learning algorithms in both systems to obtain new features.

The recommender system would propose different Hadoop configurations to improve the performance of CC applications, and the failure prediction system would propose different cases or scenarios in which a CC system could fail or simply have its performance degraded.

8.7 Summary

CC is a technology aimed at processing and storing very large amounts of data, and one of its biggest challenges is to process huge amounts of data, known as BD. In December 2012, the IDC released a report entitled, "The Digital Universe in 2020," in which the authors state that, at the end of 2012, the total amount of data generated was 2.8 ZB. As a result, BDS soon became a very important topic in organizations, because of the value it can generate, both for themselves and for their customers. However, a limiting factor in BDS is the current lack of information to help in understanding, structuring, and defining how to integrate BDS into organizations. The issues surrounding BDS integration are related to the large amounts of data from

different sources that are involved, as well as to data transformation, storage, security, etc. In this chapter, we have presented the DIPAR framework, which consists of five stages: Define, Ingest, Preprocess, Analyze, and Report. This framework proposes a means to implement BDS in organizations, and defines its requirements and elements. The DIPAR framework is based on the ISO 15939 Systems and Software Engineering—Measurement process standard, the purpose of which is to collect, analyze, and report data relating to products to be developed. In addition, we have presented the relationship between the DIPAR framework and ISO 15939. Finally, we have presented a case study which shows how to implement the DIPAR framework to create a new BD product. This BD product identifies and quantifies the *normal application behavior* of CCA. Once we had completed the implementation of the DIPAR framework, we found that the original product could be redefined to create a new product. This new product has two main functions, one as a recommender system, and another as a fault prediction system. The DIPAR framework can be implemented in different areas of BD, and we are hopeful that it will contribute to the development of new BD quality technologies.

References

1. ISO/IEC (2011) ISO/IEC JTC 1 SC38: Study Group Report on Cloud Computing, International Organization for Standardization, Geneva, Switzerland
2. Gantz J, Reinsel D (2012) The digital universe in 2020: big data, bigger digital shadows, and biggest growth in the far East, IDC: Framingham, MA, USA, p 16
3. Press GA (2013) Very short history of data science. www.forbes.com/sites/gilpress/2013/05/28/a-very-short-history-of-data-science/. Accessed May 2013
4. Tukey JW (1962) The future of data analysis. Ann Math Stat 33(1):1–67
5. Han J, Kamber M, Pei J (2012) Data mining, concepts and techniques. Elsevier, Waltham, Morgan Kaufmann, USA, 633 p
6. Lin J, Ryaboy D (2012) Scaling big data mining infrastructure: the Twitter experience. In: Goethals B (ed) Conference on knowledge discovery and data mining 2012. Association for Computing Machinery, Beijing, pp 6–19
7. Thusoo A et al (2010) Data warehousing and analytics infrastructure at Facebook. In: ACM SIGMOD international conference on the management of data 2010. Association for Computing Machinery, Indianapolis, Indiana, USA
8. ISO/IEC (2008) ISO/IEC 15939:2007 Systems and software engineering—measurement process, International Organization for Standardization, Geneva, Switzerland
9. Patil D (2012) Data Jujitsu: the art of turning data into product. O'Reilly Media, Inc., Sebastopol
10. A.F.S. (2012) Apache Flume. flume.apache.org/. Accessed 13 June 2013
11. Facebook (2012) Scribe. https://github.com/facebook/scribe/wiki. Accessed 13 June 2013
12. Kandel S et al (2012) Enterprise data analysis and visualization: an interview study. In: IEEE visual analytics science & technology (VAST), 2012, Seattle, WA, USA, IEEE Xplore
13. Thusoo A et al (2010) Hive—a petabyte scale data warehouse using Hadoop. In: 26th international conference on data engineering, 2010, Long Beach, California, USA, IEEE Xplore
14. A.S.F (2012) What is Apache Mahout? https://cwiki.apache.org/confluence/display/MAHOUT/Overview. Accessed June 2013
15. N.E.S.S.I (2012) Big data, a new world of opportunities. Networked European Software and Services Initiative, Madrid, Spain
16. Yau N (2009) Seeing your life in data. In: Segaran T, Hammerbacher J (eds) Beautiful data, the stories behind elegant data solutions. O'Reilly Media, Inc., Sebastopol, pp 1–16

17. Agrin N, Rabinowitz N (2013) Seven dirty secrets of data visualisation. February 18, 2013, www.netmagazine.com/features/seven-dirty-secrets-data-visualisation#null. Accessed June 2013
18. Coulouris G et al (2011) Distributed systems concepts and design. 5th ed. Pearson Education, Edinburgh, Addison Wesley
19. Dean J, Ghemawat S (2008) MapReduce: simplified data processing on large clusters. Commun ACM 51(1):107–113
20. Lin J, Dyer C (2010) Data-intensive text processing with MapReduce2010. University of Maryland, College Park: Manuscript of a book in the Morgan & Claypool Synthesis Lectures on Human Language Technologies
21. Xing L, Shrestha A (2005) Distributed computer systems reliability considering imperfect coverage and common-cause failures. In: 11th international conference on parallel and distributed systems, Fuduoka, Japan, IEEE Computer Society
22. A.S.F (2013) Apache HBase, the Hadoop database, a distributed, scalable, big data store. http://hbase.apache.org/. Accessed 6 June 2013
23. Rabkin A, Katz R (2010) Chukwa: a system for reliable large-scale log collection. In: Proceedings of the 24th international conference on large installation system administration, USENIX Association, San Jose, CA, pp 1–15
24. Boulon J et al (2008) Chukwa, a large-scale monitoring system. In: Cloud Computing and its Applications (CCA '08), Chicago, IL
25. Bautista L, Abran A, April A (2012) Design of a performance measurement framework for cloud computing. J Softw Eng Appl 5(2):69–75
26. Bautista L, Abran A, Abran A (2013) A methodology for identifying the relationships between performance factors for cloud computing applications. In: Zaigham M, Saqib S (eds) Software engineering frameworks for the cloud computing paradigm. Springer, London, pp 111–117
27. A.S.F (2013) Apache pig. http://pig.apache.org/. Accessed 6 June 2013