A Comparison of Predictive Analytics Solutions on Hadoop

Ramin Norousi[1], Jan Bauer[1], Ralf-Christian Härting[2], and Christopher Reichstein[2(✉)]

[1] Business Field Advanced Analytics, MHP – A Porsche Company, Ludwigsburg, Germany
ramin@norousi.de

[2] Business Administration, Aalen University of Applied Sciences, Aalen, Germany
ralf.haerting@kmu-aalen.de, christopher.reichstein@hs-aalen.de

Abstract. New approaches regarding data streaming, data storage and data analysis have been developed facing the huge volume and velocity of generated data. Enterprises are convinced that one of their key success factor is to consider available data searching for patterns and predicting the future in order to gain more insights about their business, to optimize processes and to save costs. Hence, predictive analytics has never been considered more important than it is now. Hadoop as a popular open-source framework was introduced to store and process extremely large data sets. The paper shows various ways of carrying out predictive analytics based on a Hadoop ecosystem. We investigated different solutions of both commercial vendors and open-source communities interoperating with Hadoop. Each scenario is described by its technical implementation, features and restrictions. A comparison sums up the most important issues to get a deeper insight in order to optimize Predictive Analytics Solutions based on Hadoop.

Keywords: Hadoop · Predictive analytics solutions · Big data · Spark · IBM SPSS Modeler · RapidMiner · Radoop

1 Introduction

The amount of data being collected and analyzed has been increased rapidly in the past few years. This fact caused an enormous interest in large-scale data storage and data processing. One of the initial successful approach to meet these challenges is MapReduce which was introduced in 2004 by Google [1]. MapReduce is a framework for processing parallelizable problems across huge datasets using a large number of computers (nodes), collectively referred to as a cluster. Each fragment may be executed on any node in a distributed cluster and finally the results are aggregated.

The idea of MapReduce was implemented in Hadoop [2] as an open-source framework with its underlying structure HDFS (Hadoop Distributed File System) which has become the standard for data processing of large-scale data [3, 4]. HDFS is

I. Czarnowski et al. (eds.), *Intelligent Decision Technologies 2017*,
Smart Innovation, Systems and Technologies 73, DOI 10.1007/978-3-319-59424-8_15

highly fault-tolerant and is designed to be deployed on low-cost hardware. Hadoop can be considered as a trigger that led to lot of developments in the big data community which kicked off an ecosystem of parallel data analysis tool for large clusters [5, 6]. MapReduce and its variants have been successfully applied in large-scale data-intensive applications on commodity clusters. However, these techniques were not suitable for all popular applications due to the fact that they are optimized for one-pass batch processing which make them slow for interactive data exploration. Furthermore, it was impossible for more complex, multi-pass algorithms, such as the algorithms that are common in machine learning [7]. Therefore, Apache Spark framework was proposed in 2012 by researchers at the University of Berkley [8] which overcomes these problems while allowing programmers to perform in-memory computation on large clusters. Spark as a fast and open-source engine for large-scale data processing can be considered as the next generation data processing alternative to MapReduce in the big data community [6, 9]. Furthermore Spark can outperform Hadoop up to 40 $\times$ faster than MapReduce applications, which translates directly into faster applications [10].

Based on highly successful introduced frameworks for storage of large-scale data sets, exploring and analyzing of the data is becoming more important. Thus, predictive analytics algorithms gain more and more attention in order to get insights from the large-scale data sets. Initially open-source solutions like Apache Spark were also used for analytics purposes in Hadoop by its machine learning library MLlib [11, 12]. The machine learning library of Spark as an open-source solution can be applied directly on data sets which are stored in a distributed file system like HDFS. However, it has the characteristic that it is based on scripting and coding. Hence, it is difficult to use for people with a non-programming background. Hand-coded implementation analytics can be laborious, time consuming and error-prone. Further enterprise solutions based on graphical interface can be considered as further suitable approaches which are successfully established in the analytics world. Among these, the IBM SPSS application "Modeler" and the RapidMiner application "Radoop" as leaders are investigated. Both software applications provide the possibility to build analytic models in a non-programming environment based on the data sets stored inside a Hadoop cluster [12, 13].

The paper is structured as follows: In Sect. 2, all three solutions considered in this work are described by their architecture and the process of applying analytics on Hadoop. In Sect. 3, the results based on our practical experiences with real-world data sets are summarized.

2 Distributed Analytics with Hadoop

Hadoop, as generally known, is the most suitable open-source software framework for storing and running applications on clusters of commodity hardware used by companies such as Google, Yahoo and Facebook. It provides as a massive storage for any kind of data, an enormous processing power and the ability to handle virtually concurrent jobs. Nevertheless, in order to explore the data sets on Hadoop and get accurate insights from them, there are various tools available which can be either connected to or integrated in

the Hadoop ecosystem. Among others Apache Spark (with extensions to Python), IBM SPSS Modeler and RapidMiner Radoop are considered and evaluated in this paper, which are described in more details below.

2.1 Analytics on Hadoop Using Apache Spark

Apache Spark is the most applied open-source engine for large-scale data processing and analyzing from Apache Hadoop project. Spark provides a simple and expressive programming model that supports a wide range of applications, including ETL, machine learning, stream processing, and graph computation [10]. The idea distinguishing Spark is its in-memory computation, allowing data to be cached in memory across iterations. Spark overcomes MapReduce by providing a new storage called Resilient Distributed Datasets (RDD) [7]. RDDs let users store data in memory across queries and provides seamless support by two types of applications: iterative algorithms which are common in machine learning (e.g. kernel support vector machines [13]), and interactive data mining tools that are hard to express using acyclic data flow model pioneered by MapReduce. RDDs are collections of elements partitioned across several nodes in a cluster [14–17]. Initially, it lacked a suite of robust and scalable learning algorithms until the creation of MLlib. Development of MLlib as part of the MLbase project [11] was introduced as an open-source program in 2013. Spark MLlib provides a wide range of data preprocessing, data modeling and evaluation steps on distributed data.

Architecture
The Spark engine can be integrated within the Hadoop ecosystem and consists of following four components (Fig. 1): Spark SQL as a package for working with structured data, Spark Streaming component that enables processing of live streams of data, GraphX as a library for manipulating graphs, and last but not least the Spark MLlib for machine learning approaches which is considered in this paper in more details [15]. In order to be able to manage all tasks in the Hadoop ecosystem and be efficient while maximizing the flexibility, Hadoop YARN as cluster manager is introduced.

Data Access
Apache Spark is written in Scala Programming language [18] and it runs either directly on Hadoop ecosystem or in a standalone mode. Furthermore it can read flat files and access to diverse data sources including following databases [15]:

- Distributed file systems such as HDFS, Cassandra and S3 (Amazon Simple Storage System)
- NoSQL database such as HBase
- Relational database management system such as MySQL

Additional API
The Spark machine learning library MLlib is scalable and interoperates with further programming languages. It provides a high level API to Scala, Java, Python and R. Hence, it eases the use for users which are familiar with other languages to write Spark

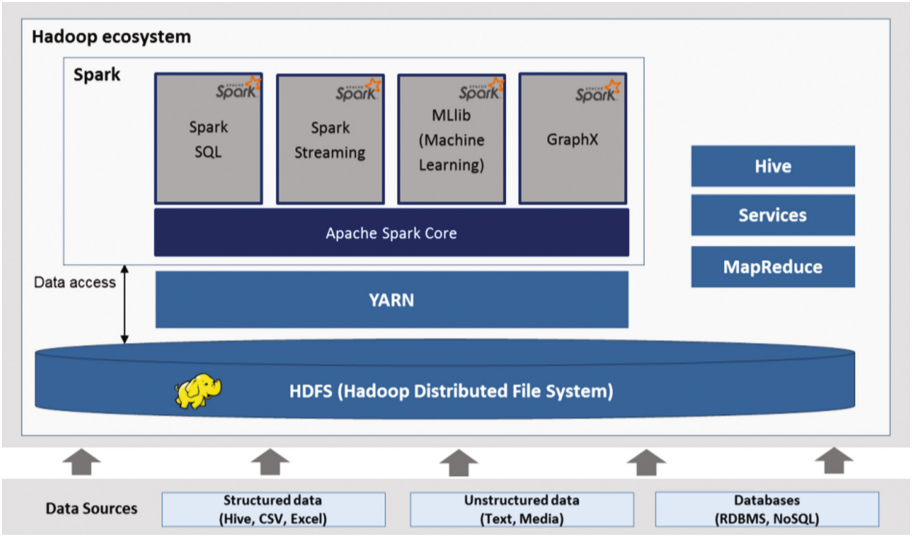

Fig. 1. Higher level architecture of Hadoop ecosystem including Spark (own source)

operations in other languages. It also offers the opportunity to access to further well-used machine learning libraries such as Numpy, Scikit-learn from Python or from R. It should be noted that in case of interoperating of Spark with further programming languages, a distributed execution is not possible.

Deployment
Due to the fact that Spark is integrated in Hadoop ecosystem, it enables to save all data preparation and predictive analytics steps in a Spark format which is able to be streamed and applied real time to Hadoop data sets. Furthermore, the MLlib supports partially model exports to the Predictive Model Markup Language (PMML), which is an XML-based interchange format to exchange models between different platform and tools [19].

Summarizing
Apache Spark is an open-source and widely-used programming model. It is integrated within the Hadoop ecosystem and it enables streaming Spark jobs in order to perform a real time execution. Furthermore, the Apache Spark community makes possible to remain constantly up-to-date regarding the analytics algorithms. Spark can also enable a distributed execution regardless of Hadoop by installing on another distributed system.

2.2 Analytics on Hadoop Using IBM SPSS Modeler and Analytic Server

IBM SPSS Modeler is a strong predictive analytics platform and one of two leaders in predictive analytics space according to a recent report from Gartner [20]. It provides a range of predictive algorithms based on a user-friendly graphical interface to support all major phases of the predictive analytics process. It has a large user base that continues

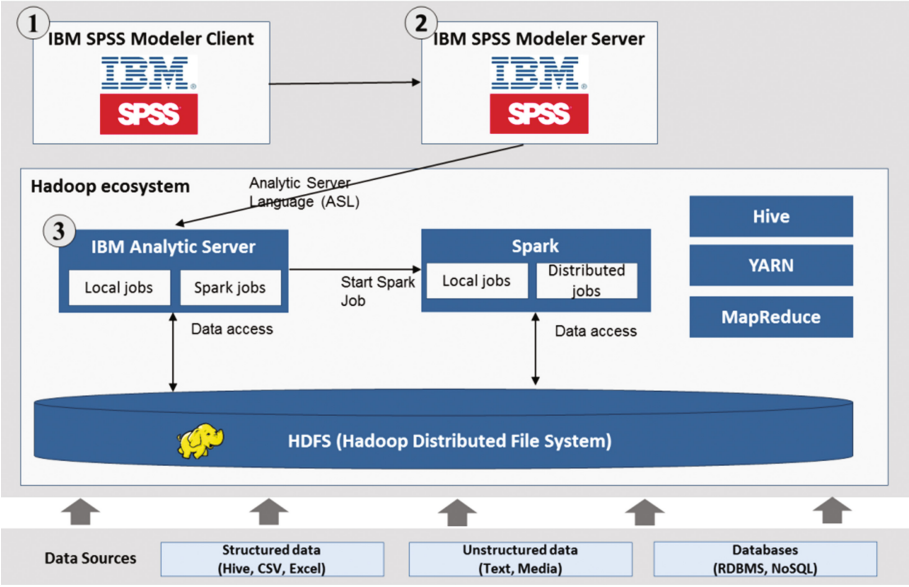

Fig. 2. Interoperating IBM SPSS and Hadoop – high level architecture (own source)

to keep up with innovation required by the market for example, integrating open-source R, Python and now Spark, to maintain high flexibility while making coding optional.

Architecture

As depicted in Fig. 2, in order to interact with data sets stored in Hadoop following three IBM components are required: IBM SPSS Modeler Client, IBM SPSS Modeler Server and IBM Analytic Server which should be initially installed as a part of the Hadoop platform and enables analysts to apply predictive analytics operations in SPSS Modeler to data stored in Hadoop. The process of performing predictive analytics jobs on Hadoop from the IBM SPSS Modeler is as follow (IBM):

1. The user develops a predictive analytics routine (called stream) on IBM SPSS Modeler Client which will be transferred into IBM SPSS Modeler Server.
2. The IBM SPSS Modeler Server receives the generated stream and translates the Stream into an IBM specific script language called Analytic Server Language (ASL) and sends it to the IBM Analytic Server which is installed on Hadoop ecosystem.
3. The IBM Analytic Server determines if the analysis should be distributed with Spark over the cluster or if it should run on the local Analytic Server JVM (Java Virtual Machine). This depends on the amount of data used for the analysis. Default settings are 128 Megabyte. Everything greater is translated into Spark Jobs and distributed over the Cluster. If spark is not available the analytic Server translates the Job into MapReduce Programs, which gets as well distributed over the cluster [21].

Data Access
Due to the fact that the IBM Analytic Server is installed directly on Hadoop, it enables to access to the same data sources like Spark:

- Distributed file systems such as HDFS, Cassandra and S3 (Amazon Simple Storage System)
- NoSQL database such as HBase
- Relational database management system such as MySQL

Additional API
IBM SPSS Modeler Client provides the option to expand the functionalities by adding algorithms which are user-written based on further programming languages. It is possible to use libraries and packages from Python and R. Furthermore, regarding to additional Hadoop services it can interoperate with Spark and use the full potential of Spark based on Analytic Server.

Deployment
All generated predictive analytics models can be stored either on the local machine or be exported with the PMML schema to other analytic tools. Furthermore, all generated predictive analytics operations in SPSS Modeler can be applied to data stored in Hadoop. Hence, IBM Analytic Server supports in-Hadoop execution of the majority of data preparation and modeling operations.

Summarizing
Providing a wide range of operations to support all major steps of predictive analytics process is an important strength of the architecture based on IBM SPSS. It can be applied on data stored in Hadoop without deep knowledge of Hadoop or Spark programming due to the graphical interface of IBM SPSS Modeler. It can access to various data bases and the major of generated operations can be applied in-Hadoop. A special feature of this architecture is that all RapidMiner operations are translated into Spark jobs first and are built on top of the MLlib in Spark. IBM SPSS Modeler is not free to use and in order to run this scenario it requires a further server for installing the IBM SPSS Server. During our practical analysis we noticed that some algorithms which are implemented are not fully distributed over the cluster, they partially run local on the Analytic Server, this may cause by the optimization engine of the Analytic Server itself.

2.3 Analytics on Hadoop Using RapidMiner Radoop

RapidMiner Radoop is a code-free analytics solution for Hadoop which has no separate service on Hadoop. RapidMiner Radoop is a fully graphical tool supporting the whole range of data analytics from ETL and ad-hoc reporting to predictive analytics [22].

Architecture
As it is depicted, RapidMiner Radoop does not require any installation on Hadoop. It offers two possibilities for analyzing and visualizing large-scale data sets stored in Hadoop:

1. RapidMiner client with Radoop as a client application that simplifies creating, maintaining and running analytics jobs over Hadoop directly.
2. RapidMiner server with Radoop as collaboration, scheduling, web reporting and web service integration to make it easier deploying big data analytics processes into an existing enterprise environment. It is a great choice for companies with a large Hadoop cluster and many users who wish to analyze and visualize big data.

In order to access the data on Hadoop, all operations developed on RapidMiner are translated into Spark or Hive jobs and transmitted to the Hadoop ecosystem. Hive is structuring the data for further analysis. It uses concepts like tables or columns to present the data. It translates SQL familiar Queries into MapReduce and HDFS tasks [23] (Fig. 3).

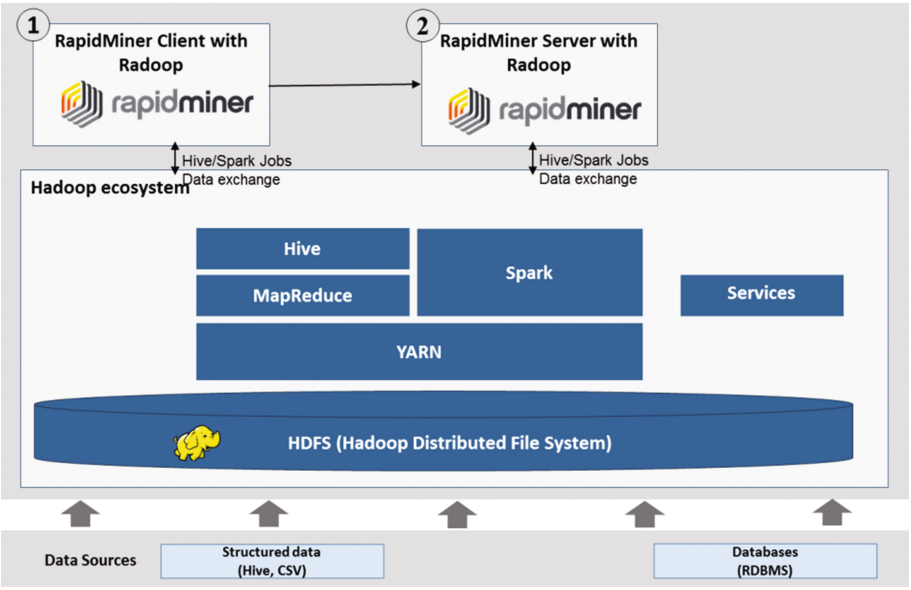

Fig. 3. Interoperating RadipMiner and Hadoop – high level architecture (own source)

Data Access

As RapidMiner Radoop runs all data preprocessing steps in Hive, data must be converted in Hive to use it with RapidMiner Radoop. It is possible to convert CSV-Files directly into Hive Tables and to access different Data sources which are listed below:

– Distributed file systems such as HDFS, Cassandra and Amazon Simple Storage (S3)
– Relational database management system such as MySQL

Additional API
It is possible to expand the RapidMiner functionalities by adding user-written operations based on further programming languages like Python or R.

Regarding to additional Hadoop services, it can interoperate with Spark and use the full potential of Spark based on Analytic Server. It can also interact with Hive and Pig.

Deployment
Predictive analytics operations which are built with a Spark script can be saved within spark itself. Furthermore, it is possible to use the generated models either outside of Hadoop with RapidMiner itself on local machine or on other analytic tools based on PMML-export. It supports in-Hadoop execution the essential of data preparation and modeling operations.

Summarizing
The architecture based on RapidMiner has also the advantage that it provides a wide range of operations to support all steps of a predictive analytics process. It can be applied on data stored in Hadoop without deep knowledge of Hadoop or Spark programming due to the graphical interface of RapidMiner. Furthermore, it can access various data bases and the major of general data analysis operations can be applied in-Hadoop.

A special feature of this architecture is the fact that all RapidMiner operations are translated into Spark jobs first and are built on top of the MLlib in Spark. It can interoperate additionally with further Hadoop services like Hive and Pig. The native RapidMiner is a free of use software. The version including Radoop in order to connect a Predictive Analytic application with Hadoop is covered by licence fees. Furthermore, this scenario can run without an additional server component for RapidMiner Server.

3 Results of Applying Analytics on Hadoop

Following approaches were considered based on a large real-world data set:

- Spark as an open-source cluster computing framework from Apache. It performs in-memory computation on large data sets comprising MLlib (Machine Learning library).
- IBM as a leader vendor with a high visibility in the advanced analytics space providing its strong product IBM SPSS Modeler with a graphical user interface and a wide range of analytics functionalities.
- RapidMiner as a further leader of analytics vendor. It provides a basic and community editions which is free and open-source and a commercial professional edition Radoop that has the ability to work with Hadoop.

According to our practical implementation and test based on three mentioned scenarios, findings are summarized in this section. The table below shows different aspects of the

Table 1. Solutions for data analytics on Hadoop

Criterion	Spark	IBM SPSS Modeler & Analytic Server	RapidMiner Radoop
Data Access	Structured (CSV) Unstructured NoSQL RDBMS Hive	Structured (CSV) Unstructured NoSQL RDBMS Hive	Structured (CSV) RDBMS Hive
Additional Hadoop services interaction	Hive	Spark written in Python	Spark written in Python or R Hive Pig
Deployment	In-Hadoop execution, export with PMML	In-Hadoop execution, export with PMML	In-Hadoop execution, export with PMML
Open-source	Yes	No	No
Independent of Hadoop	Yes	No	No
Community support	Yes	No	No
Hand Coded/GUI	Hand Coded	GUI/Hand Coded	GUI/Hand Coded
Data export	PMML Data Export to local or HDFS	PMML Data Export to local or HDFS	PMML Data Export to local or HDFS

solutions compared to each other. The main parts are highlighted. It should help to gain more knowledge about the different solutions for data analytics on Hadoop (Table 1).

Furthermore, available predictive analytics algorithms in each scenario can be evaluated (Table 2).

Each predictive analytics solution interoperating with Hadoop has its strengths and weaknesses. The choice of the appropriate solution depends on each specific application. It should be determined on how many data sources should be read, how complex the data preparation steps are, which predictive analytics algorithms should be applied and finally the generated models be deployed and integrated in-Hadoop or on other databases.

To sum up, based on five selected criteria a recommendation can be made according to the three solutions as shown in Table 3. The selected criteria can be considered as main steps of a data mining process. Based on a real data set, we tested the performance and number of available operations of all three analytics solutions according to these criteria. All three scenarios are assessed based on defined facts to make recommendations from our tests.

Table 2. Predictive analytics algorithms in each scenario

Methods	Algorithms	Spark	SPSS Analytic Server	RapidMiner Radoop
Classification	Decision Tree based on Gini-Impurity	X	X	X
	Decision Tree based on Entropy	X	X	X
	Decision Tree based on Chi-square independence		X	
	SVM (Linear)	X	X	X
	Naive Bayes	X		X
	KNN			
	Neural Network	X	X	
Regression	Linear Regression	X	X	X
	Logistic Regression	X	X	X
Clustering	K-Means	X		X
	TwoStep		X	
Association Rules	FP-Gwoth	X		
	Association Rules	X	X	
	Apriori			
Others	PCA	X		X
	Time Series Analysis		X	
Ensemble	Random Forest	X	X	X

Table 3. Recommendation based on five selected criteria

Criterion	Recommendation	Facts
Data Access	Spark, IBM SPSS	Accessing to five different data sources
Data Preparation	RapidMiner, IBM SPSS	Number of available operations
Modeling	All three solution	Number of available algorithms
Evaluation	IBM SPSS Modeler, RapidMiner	Number of available statistics for error measurements
Deployment	Spark	Additionally to in-Hadoop execution, it provides streaming of generated models

4 Conclusion

The number of unexpended data is growing at a breathtaking pace year by year. Scientists as well as managers have to deal with an exponential growing number of data in the next years [24]. In order to provide a remedy, there are different solutions in the market place both commercial and open-source. This paper compared three different and widely used Predictive Analytics Solutions on Hadoop based on a large real-world data set: Spark, IBM and RapidMiner. First, we investigated each solution by itself

describing its technical implementation, features and restrictions. Afterwards, we considered Predictive Analytics Algorithms in different scenarios and compared all three solutions with each other in terms of data access, data preparation, modeling, evaluation and deployment. As a result, the choice of an appropriate Predictive Analytics Solution mainly depends on its specific application, the volume of data that should be read and the complexity of the data preparation. For firms with less complex data structures, classical data management concepts (i.e. on-premises data warehouse solutions) within the company might be sufficient.

The results contribute to the current research. Using Predictive Analytics Algorithms helps to understand how these solutions work within different scenarios. A comparison shows the strengths and weaknesses of each Predictive Analytics solution to reveal potential opportunities and risks when operating with Big Data [24].

Beyond, there are also managerial implications regarding Predictive Analytics Solutions. Managers can use our study as a recommendation when and which Predictive Analytics Solution is to use. An appropriate usage with the most convenient Predictive Analytics Solution might help to better understand the mass of data. Thus, analyzed data about customers might be useful to get essential insights about customer needs which in consequence lead to higher customer satisfactions and higher profits.

There are also some limitations regarding our examination. First, there are different aspects that were not taken into consideration like company size, different usage in different industries as well as different technical know-how within a company and/or possible missing IT infrastructures. Especially the last aspect is an important issue for managing Big Data in "the age of cloud computing" [5, 6], which is connected with the choice between Cloud- or On-Premises-Hadoop-Technologies. Second, we only considered three different Predictive Analytics Solutions. In fact, there are many more services available to analyze Big Data. Some of these services are based on new concepts. So called Data Lakes are integrating classical Data Warehouses and Hadoop-Clusters. Examples are HDInsight, a Microsoft Hadoop-Platform, based on Hortonworks Data Platform (HDP) or Analytics Platform System (APS).

Hence, future research should focus on investigation of various Predictive Analytics Solutions within firms depending on different industries, levels of know-how and IT infrastructures to gain deeper knowledge about an optimized use of data analytics.

References

1. Dean, J., Ghemawat, S.: MapReduce: simplified data processing on large clusters. In: Proceedings of the 6th Conference on Operating Systems Design and Implementation (OSDI), p. 10. USENIX Association, Berkeley (2004)
2. White, T.E.: Hadoop: The Definitive Guide, 3rd edn. O'Reilly, Sebastopol (2012)
3. Shvachko, K., Kuang, H., Radia, S., Chansler, R.: The Hadoop distributed file system. In: Shvachko, K., Kuang, H., Radia, S. (eds.) 26th Symposium on Mass Storage Systems and Technologies (MSST), pp. 1–2. IEEE, Incline Village (2010)
4. Zhao, J., Wang, L., Tao, J., Chen, J., Sun, W., Ranjan, R., Georgakopoulos, D.: A security framework in G-Hadoop for big data computing across distributed Cloud data centres. J. Comput. Syst. Sci. **80**(5), 994–1007 (2014)

5. McAfee, A., Brynjolfsson, E., Davenport, T.H., Patil, D.J., Barton, D.: Big data. The management revolution. Harv. Bus. Rev. **90**(10), 61–67 (2012)
6. Hashem, I.A.T., Yaqoob, I., Anuar, N.B., Mokhtar, S., Gani, A., Khan, S.U.: The rise of "big data" on cloud computing: review and open research issues. Inf. Syst. **47**, 98–115 (2015)
7. Zaharia, M., Chowdhury, M., Franklin, M.J., Shenker, S., Stoica, I.S.: Cluster computing with working sets. In: Proceedings of the 2nd USENIX Conference on Hot Topics in Cloud Computing (HotCloud), p. 10. USIENIX Association, Berkeley (2010)
8. Srirama, S.N., Jakovits, P., Vainikko, E.: Adapting scientific computing problems to clouds using MapReduce. Future Gener. Comput. Syst. **28**(1), 184–192 (2012)
9. Sagiroglu, S., Sinanc, D.: Big data: a review. In: International Conference on Collaboration Technologies and Systems (CTS), pp. 42–47. IEEE, San Diego (2013)
10. Zaharia, M., Chowdhury, M., Das, T., Dave, A., Ma, J., McCauley, M.: Resilient Distributed Datasets: A Fault-Tolerant Abstraction for In-Memory Cluster Computing. EECS Department, University of California, Berkeley (2011)
11. Meng, X., Bradley, J., Yavuz, B., Sparks, E., Venkataraman, S., Liu, D.: MLlib: machine learning in apache spark. J. Mach. Learn. Res. **17**(34), 1–7 (2016)
12. Zikopoulos, P., Eaton, C.: Understanding Big Data: Analytics for Enterprise Class Hadoop and Streaming Data. McGraw-Hill Osborne Media, New York (2011)
13. Patel, A.B., Birla, M., Nair, U.: Addressing big data problem using Hadoop and MapReduce. In: Nirma University International Conference on Engineering (NUiCONE), pp. 1–5. IEEE, Ahmedabad (2012)
14. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification, 2nd edn. Wiley-Interscience, New York (2012)
15. Apache Spark: Apache Spark™ - Lightning-Fast Cluster Computing. https://spark.apache.org/. Accessed 11 Jan 2017
16. Wu, X., Zhu, X., Wu, G.Q., Ding, W.: Data mining with big data. IEEE Trans. Knowl. Data Eng. **26**(1), 97–107 (2014)
17. Witten, I.H., Frank, E., Hall, M.A., Pal, C.J.: Data Mining: Practical Machine Learning Tools and Techniques. Morgan Kaufmann, Burlington (2016)
18. Odersky, M., Venners, B., Spoon, L.: Programming in Scala, 2nd edn. Artima Press, Walnut Creek (2011)
19. DMG: Data Mining Group. http://dmg.org/. Accessed 17 Jan 2017
20. Kart, L., Herschel, G., Linden, A., Hare, J.: Magic quadrant for advanced analytics platforms. Gartner report 9 (2016)
21. IBM: IBM SPSS Analytic Server Version 3.0: Overview. ftp://public.dhe.ibm.com/software/analytics/spss/documentation/analyticserver/3.0/English/IBM_SPSS_Analytic_Server_3.0_Overview.pdf. Accessed 19 Jan 2017
22. RapidMiner Radoop: RapidMiner Radoop - RapidMiner Documentation. http://docs.rapidminer.com/radoop/. Accessed 19 Jan 2017
23. Thusoo, A., Sarma, J.S., Jain, N., Shao, Z., Chakka, P., Zhang, N.: Hive - a petabyte scale data warehouse using Hadoop. In: IEEE 26th International Conference on Data Engineering (ICDE), pp. 996–1005. IEEE, Piscataway (2010)
24. Fan, W., Bifet, A.: Mining big data: current status, and forecast to the future. ACM SIGKDD Explor. Newsl. **14**(2), 1–5 (2013)