Hindawi Publishing Corporation
Advances in Agriculture
Volume 2016, Article ID 7081491, 6 pages
http://dx.doi.org/10.1155/2016/7081491

Research Article

Analysis of Plant Breeding on Hadoop and Spark

Shuangxi Chen,[1] Chunming Wu,[2] and Yongmao Yu[2]

[1]*Jiaxing Vocational Technical College, No. 547 Tongxiang Road, Jiaxing, Zhejiang 314036, China*
[2]*Zhejiang University, No. 38 Zhejiang University Road Yuquan Campus, Hangzhou 310012, China*

Correspondence should be addressed to Shuangxi Chen; 305468248@qq.com

Received 7 December 2015; Revised 4 April 2016; Accepted 11 April 2016

Academic Editor: Tibor Janda

Analysis of crop breeding technology is one of the important means of computer-assisted breeding techniques which have huge data, high dimensions, and a lot of unstructured data. We propose a crop breeding data analysis platform on Spark. The platform consists of Hadoop distributed file system (HDFS) and cluster based on memory iterative components. With this cluster, we achieve crop breeding large data analysis tasks in parallel through API provided by Spark. By experiments and tests of Indica and Japonica rice traits, plant breeding analysis platform can significantly improve the breeding of big data analysis speed, reducing the workload of concurrent programming.

1. Introduction

Crop breeding technology is an important means of increasing crop yields; it also has important strategy on social stability and sustainable development [1]. With the rapid development of computer technology, crop breeding has efficient and accurate data analysis and progeny selection on computer. On the basis of qualitative description of the crop breeding, the key technology is a combination of molecular biology, physiology, and the characteristics of precise quantification. Especially with the rise of biometrics, automated sampling, and digital technology, there is a great leap on the molecular breeding. We utilize the use of Hadoop distributed cloud storage technology as data storage, using Apache Spark as large-scale data processing engine. Based on big data analysis capabilities, we provide universal high cluster-computing platform for crop breeding large data analysis and processing, with the combination of data mining algorithm package MLlib and real-time streaming data processing. The platforms integrate with one another to be able to give in full play its advantages in resources to level of gene expression and the macro performance data [2].

With the development of the Internet and the rise of cloud computing, big data analytics technology has become an important trend in the development and application of a large amount of data mining. Numerous applications emerge like mushrooms after rain. For example, doctors can rationally use drugs according to data analysis on tumor DNA and normal DNA. Besides, through the rainfall over the past few decades in a region, the time when water rises can be predicted [3]. Therefore, considering the current situation of Chinese breeding, breeding technology is put forward based on analysis of big data.

2. Breeding Problem

Sustainable innovation of modern science and technology leads to profound changes in crop breeding. With speedy development, new technologies are applied, involving bionics, biotechnology, information technology, and manufacturing technology. When science and technology continuously infiltrate different aspects, they give birth to a novel crop breeding system. Meanwhile, the data obtained presents a sharp increase in crop breeding. Meanwhile, the data obtained presents a sharp increase in crop breeding. Other than field survey, our experimental data originates from gene data (e.g., gene expression and molecular marker), metabolites dynamic data, production and management data, and dynamic environmental data (e.g., moisture, air temperature, and water content). Teng defined digital breeding, as "Through standardized management and analysis of extensive dynamic breeding data, synthesized attributes

of breeding materials will be automatically processed After genetic distance and groups analysis of breeding material, we can determine hybrid vigor in advance. Taking environmental factors and field trials into consideration, breeding results can be selected on-demand [4]". Since heterogeneous and heterologous breeding data increases dramatically, big data seems ready to come out for better improving accuracy and efficiency of crop breeding [5].

Currently, breeding-related data include internal data, documents, and other data related to genetic resources. These data are pretty large, but scattered, without effective organization. Currently, breeders in the breeding process consider data mainly from its internal data, not much from the literature and genome-wide association [6]. These problems greatly limit the use and improvement of breeding efficiency.

In this paper, we propose data storage solutions based on breeding Hadoop platform, combined with Spark platform iterative calculation, while ensuring high fault tolerance breeding data and high performance storage, making massive breeding data analysis done quickly.

3. System Framework

As Apache's Hadoop distributed application development framework, Hadoop distributed file system is one of the popular cloud storage platforms. Map Reduce programming paradigm and HDFS (Hadoop distributed file system) are the core Hadoop framework of the two technologies. HDFS is a streaming data access pattern and can store large distributed file management system that uses write once, read many models. HDFS with its high reliability and high performance characteristics is especially suitable for deployment in the commercial computer consisting of a cluster [7].

Spark is an open source cluster-computing framework originally developed in the AMPLab at UC Berkeley. By allowing user programs to load data into a cluster's memory and query it repeatedly, Spark is becoming the core technology of big data and cloud computing. Spark project integrates the SparkSQL, Spark Streaming technology, solving the big data in batch, streaming, ad hoc queries, and other three core issues [8].

As shown in Figure 1, the computer nodes are constituted by a plurality of computer nodes clustered environment, which is on the lowest level breeding analysis platform, on computer clusters, building Hadoop distributed file system (HDFS) provides high performance and high fault-tolerant distributed file data. The file system is stored not only for breeding analysis platform data centers, but also for the Apache Spark analysis platform and other components provided with real-time data persistent storage and analysis of data collection.

As shown in Figure 2, breeding data center is the core of the whole framework, providing breeding data analysis services, real-time breeding data stream processing, and offline batch processing. The specific functions are as follows.

(1) Heterogeneous Data Acquisition System. As a big data breeding analysis platform, it is needed to integrate the different data sources, such as crop lines, genes, traits, and other

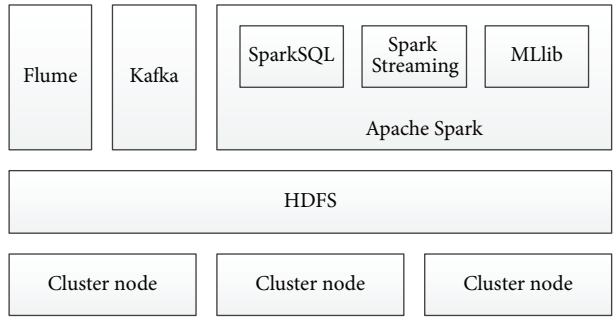

Figure 1: The system architecture of breeding.

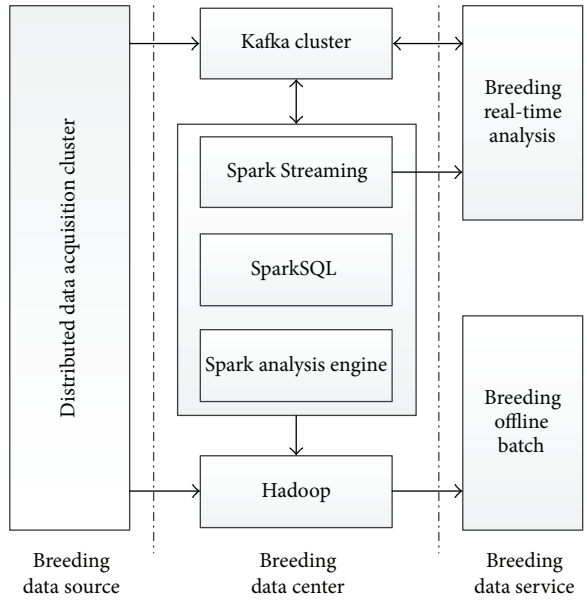

Figure 2: Breeding data flow diagram.

information management system data. Cloudera Flume as a highly available, highly reliable distributed data acquisition system, its log system supports a variety of sources, such as ordinary files, and transfer of TCP network, providing a data interface for breeding analysis platform [9].

(2) Breeding Data Message Subscription and Publishing System. Breeding data traffic generated by the face of the data acquisition system, the data acquisition speed, and data processing speed may not be synchronized, because of its volatile characteristics of streaming data, such as the ability not only to analyze but also to process platform will result in data loss. Kafka as a distributed messaging system supports sending message, subscription, and the message stored in local files [10]. Faced with the real-time breeding data streams, Flume achieves data generated by breeders and breeding data analysis platform decoupling consumers to protect breeding high reliability and consistency of the data. In addition, Kafka achieves buffered data streams through subscription and news release mode, processing tasks in order to prevent clogging of breeding analysis platform. Spark Streaming is a real-time data streaming component that

supports multiple data sources such as Kafka and Flume with the Spark analysis platform to implement distributed data streams for breeding Kafka processing.

(3) Distributed Data Warehouse. Data Warehouse as a subject-oriented data collection, such as rice, maize germplasm bank, and genetic resource base as the theme of the data warehouse, typically provides structured data collection for breeding data analysis. SparkSQL as Spark components provides similar structured query language (SQL) for relational databases, through the system to get from Kafka breeding information stream into a database table which is similar to distributed datasets (SechemaRDD). Compared to traditional data warehouse, SparkSQL combined with Spark and Hadoop HDFS platform has distributed data processing capability and distributed file storage capacity [11]. In addition, SparkSQL provides a common interface to other data management systems such as MySQL and HIVE (Hadoop-based distributed data warehouse), to provide a data source for other breeder's heterogeneous platforms.

(4) Breeding Data Analysis Library. In the process of breeding analysis, it often involves mathematical statistics and data mining, machine learning, and other applications. MLlib as the achievement of machine learning algorithm on Spark platform supports common machine learning problems such as classification, regression, clustering, and collaborative filtering. With the integration of Kafka and Spark Streaming, the MLlib provides real-time analysis of large datasets and offline batch supports distributed analysis algorithm for breeding analysis, which reduces development effort.

4. Core Algorithms

Crop breeding analysis uses the existing biometric methods and quantitative genetic analysis models tested strains traits comprehensive evaluation and analysis decision breeding material, by means of data mining, machine learning haphazard breeding data from a data mining law, construction breeding analysis model, and then guide the breeding to speed up the process.

MLlib is an achievement of machine learning algorithm on Spark analysis platform. MLlib supports four types of common machine learning problems: a binary classification, regression, clustering, and collaborative filtering [12]. As a supervised learning problem, classification algorithms currently support MLlib linear SVM (support vector machine) and logistic regression. Regression algorithm contains a linear regression and the associated L1 (lasso) and L2 (Ridge) regularization variant, which is commonly used in the forecast, such as field trials strain effect estimates and projections in crop breeding. These three types of underlying algorithms are called gradient descent optimization algorithm provided by MLlib. Clustering algorithms are often used for exploratory analysis experiments and also widely used in crop breeding to reflect the differences in the genome of genomes by the distance to measure points. The clustering algorithm of MLlib provides Universal K-means algorithm, which is according to the number of user-defined algorithm

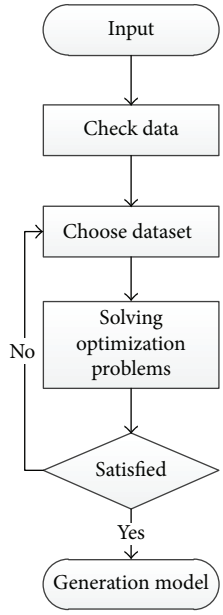

Figure 3: Classification algorithm flowchart.

to cluster. Collaborative filtering algorithm is typically used in recommendation system. MLlib provides collaborative filtering algorithm based on the model, with the implicit semantic factor to express merchandise utility user matrix.

As shown in Figure 3, there are a lot of dataset pretreatment methods in MLlib algorithm library, such as breeding dataset provided by MLUtil [13].

Data validators check the training set and count the number of success, generating breeding training set. In machine learning algorithm, the most common method is by gradient descent algorithm to minimize the expected risk through calculating the loss function minimum mean of predicting actual value to select the optimizing model.

MLlib provides a SGD (Stochastic Gradient Descent) sequence optimization algorithm to solve optimization problems. In the process provided by gradient descent class, you can set the number of steps to initialize SGD. SGD iteration datasets fragment size, SGD iterations, and so forth control the SGD algorithm iterative process. When it reaches SGD algorithm's requirements, it stops and outputs optimization model. In the actual breeding process, in addition to use of the existing machine learning methods, we can quickly build breeding analysis model through SGD. MLlib model also provides an assessment of the quality indicators of methods, such as model predictive accuracy and the recall rate.

5. The Implementation of Breeding

Big data breeding system architecture is divided into 5 layers as below.

(1) Data Interactive Interface Layer. Through Sqoop tool, database established in Hadoop distributed file system (HDFS) can interact and synchronize with relational database.

Table 1: Cluster configuration information table.

Name	Model	Number	Configuration
Main server	Dell server	1	CUPE5-2609 master server quad-core/16 G memory/500 G SATA hard drive
Sub server	Dell server	3	CUPE5-2609 master server quad-core/16 G memory/500 G SATA hard drive

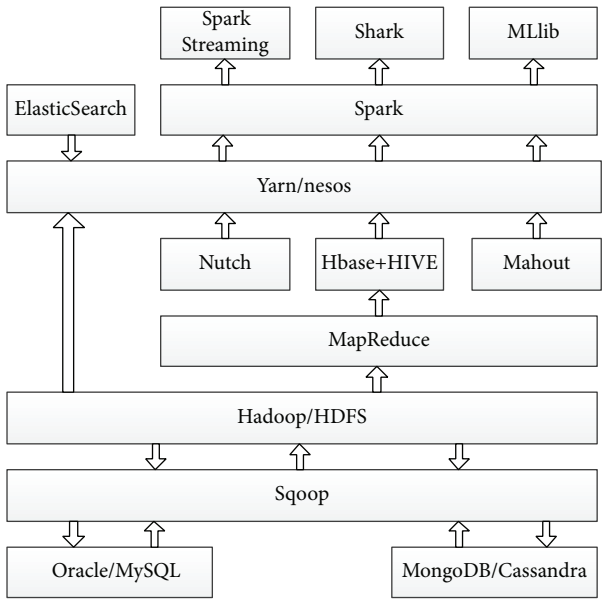

Figure 4: System structure diagram.

(2) Hadoop Analysis Platform Layer. In view of mass offline breeding data, data mining batch allocation is achieved. Furthermore, when combining with Mahout data mining algorithm package in Hadoop, breeding data can be rapidly modeled.

(3) Yarn Resource Management Layer. In this layer, resource management is unified in Hadoop cluster, with combination of Spark on yarn running mode and Spark analysis platform.

(4) Spark Analysis Platform Layer. Real-time analysis platform is constructed to satisfy iterative computation scenario. On the basis of memory iteration data mining algorithm package in Spark, MLlib can be integrated to build data mining model quickly.

(5) Vertical Search Layer. Together with web crawler Nutch and distributed database Hbase, full-text index is established to provide full-text search. System architecture is shown in Figure 4.

Big data breeding function module mainly contains the following aspects.

(1) Data Import and Export. Traditional database RDBMS and NoSQL database can interact and synchronize with HDFS in Hadoop framework through Sqoop. For instance, data from MySQL, Oracle, and Postgres can be imported into HDFS, and vice versa.

(2) Hadoop Cluster Management and Monitoring Platform. Via cluster management platform, breeding analysis platform can timely monitor nodes in Hadoop cluster, including hot plug nodes, extendable nodes, intelligent nodes, message and e-mail alerts, smart diagnosis and correction of node failure, graphical user interface, and drag and management of HDFS, which reduces the difficulty of managing and monitoring Hadoop cluster.

(3) Data Warehouse Platform. Based on structured and unstructured mass data storage, ETL tool is applied to implement distributed data cleaning, transformation, and loading, and thus a topic model data warehouse is created.

(4) Hadoop Analysis Platform. Hadoop analysis platform contains HDFS, MapReduce operation, column-oriented database Hbase, Distributed ETL, data warehouse HIVE, and Mahout algorithm package based on MR algorithm. In addition, this platform supports mass data source required for data mining, data cleaning, a series of ETL operations (e.g., data extraction, data transformation, data loading), and distributed data mining algorithm, such as classification, clustering, regression forecasting, and association rules.

(5) Spark Analysis Platform. Spark distributed cluster is established in Hadoop framework, mainly containing cluster management, data warehouse shark, stream-based algorithm package Spark Streaming, and data mining package MLlib applied in iterative computation scenario which includes common models like classification, clustering, and regression forecasting. This allows for real-time analysis of breeding data.

6. Analyses and Performance Testing

6.1. Experimental Environment. We use Apache Spark system for Indica rice and Japonica traits dataset to predict the course of Indica or Japonica by data mining algorithms and modeling to test single cluster contrast to the same dataset amount of time spent modeling process.

As shown in Table 1, Hadoop cluster consists of three machine components, in which one server acts as the Hadoop cluster nodes and the Spark Master node in the cluster. NameNode node set Master node does not participate in file storage and computing nodes. As in the other two servers DataNode framework Hadoop cluster nodes and Spark worker nodes in the cluster, NameNode is responsible for file storage, and worker bear node computing tasks. The entire cluster has 16 cores, 64 G memory, 2 T disk space. As shown in Table 2, software and corresponding version are listed, which are used in the experiment. As shown in Table 3, the Hadoop cluster configuration is listed, which is used in the experiment.

Table 2: Experimental software version table.

Name	Version
Linux	Ubuntu 12.04.5
DK	jdk-7u67-linux-x64
Nutch	Apahce-Nutch-2.1
Hadoop	hadoop-2.2.0
Spark	spark-1.0.0-bin-hadoop2
ANT	ANT-1.9
Scala	scala-2.10.4
ElasticSearch	Elasticsearach-1.4

Table 3: Hadoop cluster configuration table.

Cluster node attribute	Machine name	IP
Master	ngnt-R720-2	10.15.123.111
Slave	ngnt-R720-3	10.15.123.112
Slave	ngnt-R720-4	10.15.123.113
Slave	ngnt-R420-A	10.15.123.114
Slave	ngnt-R420-B	10.15.123.115

6.2. Data Preparation. Experimental data description of Indica or Japonica trait dataset is provided by the Chinese National Germplasm Repository. A total of 5,3 000 datasets of data, a total of 55 property fields, and part of the data attribute fields are incomplete. According to rice subspecies Indica property characteristics, especially the selection of describing important features of Indica or Japonica subspecies and other property items, we remove items missing data, in which the total number is more than 12,000 dataset. Part of the selecting data is shown in Table 4.

In the experiment, in order to meet the cluster performance testing, data collection through its Indica rice japonica accounting ratios copy thousand times, so that the model training dataset reached 15 million, the total size of the dataset storage space is 1 G.

Note that the sharing file's URL of the raw data of rice breeding is http://pan.baidu.com/s/1miRGaso. The sharing file's URL of the cleaned data is http://pan.baidu.com/s/1o8CcTia. The sharing file's URL of the SVM model input data is http://pan.baidu.com/s/1gfLeBb9.

6.3. Test and Performance Testing. In Spark, we wrote Spark application, by reading the Hadoop HDFS rice datasets generating Spark RDD, calling the SVM model algorithm provided by MLlib, and loading required datasets. In the model training process, by datasets cross-validation method, the datasets were randomly split into a training set and test set, the ratio of 3 : 1, and then train the model. In the whole training process model, the model can be monitored through the training process and the training time by Spark job monitoring system.

As shown in Figure 5, the model training process adopts memory iterative algorithm. In comparison with compute nodes, a model training time is significantly reduced, which shows the memory effect is based on iterative calculations significantly.

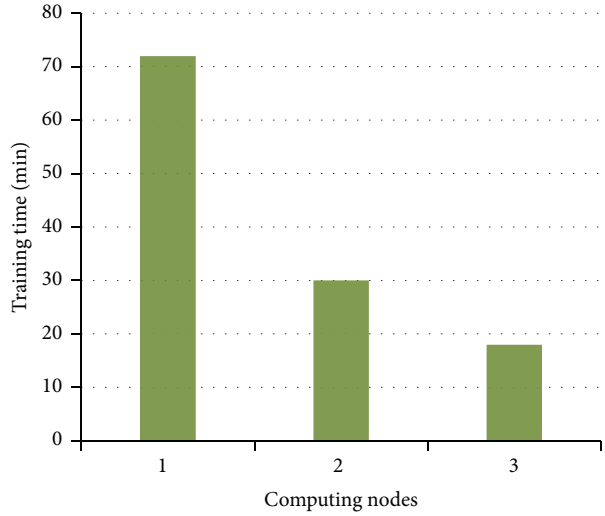

Figure 5: Model training time comparison chart.

In binary classification problem of model checking process, the instance of the class is divided into positive and negative categories. This is a similar case with prediction of negative class, including false negative category and false positive type. Calculated by the receiver operating characteristic (ROC) curve and AUR (area under ROC), model train measure computes nodes according to correct rate and recall rate. In this experiment, the resulting cross-validation model AUR area under the curve is 0.895, close to the 0.9. According to the AUR criteria, if AUR value reached 0.9, this indicates that the model has high prediction accuracy.

Note that the sharing file's URL of the adjustment of training parameters and AUC value of SVM model is http://pan.baidu.com/s/1jI6LwDS8. The sharing file's URL of the Scala source code based on Spark1.3 and running on cluster is http://pan.baidu.com/s/1boMs111. The sharing file's URL of the Scala source code based on Spark1.3 and running on local is http://pan.baidu.com/s/1pKJSgMR. The sharing file's URL of the experimental results is http://pan.baidu.com/s/1qYcgXcC.

In addition, due to the presence of Chinese character in the files, all of the shared file is encoded by UTF-8.

7. Conclusion

Based on real-time analysis of Hadoop Spark platform, Hadoop generally acts as mass data storage system and offline batch analysis scenarios. Spark memory-based iteration characteristics provide a wealth of big data processing and analysis components to meet the real-time data analysis breeding, offline analysis, and other multiscene application requirements. The next study is to develop analytical platform on top of the original set of breeding analysis platform management systems, build plant breeding analysis process through the workflow engine, add the breeding cycle of executive function tasks, and implement breeding process analysis tasks, the

Table 4: Traits of Indica or Japonica dataset tables.

Rice	Heading stage	Height	BRR	MRR	Protein	Lysine	Starch	Amylose	Amylopectin	Pasting temperature	Gel consistency
Japonica	8.22	110	77.2	69.8	11.68	0.406	74.98	26.9	56.4	6.5	60
Japonica	8.17	127	78.6	70.4	11.86	0.411	73.89	25.5	56.6	7	95
Japonica	8.28	105	79.5	71.9	10.6	0.367	75.91	26.4	57.9	6	85
Indica	8.3	148	76.2	68.2	8.81	0.336	76.84	21.8	63.6	6	49

front page controls achieve breeding analysis process visualization management. In addition, breeding algorithm and breeding data source can be configured. Moreover, dynamic binding data set management algorithms show visualization and data visualization pluggable components. Therefore, real breeding analysis platform becomes a breeding process control, understandable large data management, and service platform.

Competing Interests

The authors declare that they have no competing interests.

Acknowledgments

This work was supported by the foundation of the National Science and Technology Support Program, ID 2014BAH24F01; National Science and Technology Platform; National Crop Germplasm Resources Platform (subplatform: National Crop Germplasm Resources Data Organization model NICGR2015-075); Public Projects of Zhejiang, ID 2016C31096; Jiaxing Science and Technology Project, ID 2014AY21021.

References

[1] H. Chen, W. Zhang, and L. Fan, "Methodology of crop breeding: progress and prospect," *Bulletin of Science and Technology*, vol. 27, no. 1, pp. 61–63, 2011.

[2] D. Chun-shui and C. Zhuo, "Advances in modern data-driven breeding technologies," *Journal of Maize Sciences*, vol. 21, no. 1–8, pp. 1–2, 2013.

[3] T. M. Li, J. Y. Chen, and D. D. Yan, "Analysis of application prospect of big data," in *Proceedings of the Academic Annual Conference of Sichuan Communication Association*, pp. 67–69, 2014.

[4] H. T. Teng, "Exploration on digital maize breeding," *Chinese Agricultural Science Bulletin*, vol. 12, no. 24, pp. 495–498, 2008.

[5] L. J. Fang, W. D. Wang, B. Wang, C. Y. Ye, Q. Y. Shu, and H. Zhang, "Crop breeding-related data and application of big data technologies in crop breeding," *Journal of Zhejiang University (Agriculture & Life Sciences)*, vol. 42, no. 1, pp. 30–39, 2016.

[6] D. Zhu, C. Wang, X. Wang, C. Yu, and C. Zhao, "Application of information technology in crop breeding," *China Rice*, vol. 17, no. 6, pp. 25–28, 2011.

[7] C. Lam, *Hadoop in Action*, Manning Publications, 2010.

[8] M. Zaharia, T. Das, H. Li, S. Shenker, and I. Stoica, "Discretized streams: an efficient and fault-tolerant model for stream processing on large clusters," in *Proceedings of the 4th USENIX conference on Hot Topics in Cloud Computing (HotCloud '12)*, Boston, Mass, USA, June 2012.

[9] U. Han and J. Ahn, "Dynamic load balancing method for apache flume log processing," *Advanced Science and Technology Letters*, vol. 79, pp. 83–86, 2014.

[10] N. Garg and A. Kafka, *Birmingham B3 2PB*, Packt Publishing, Birmingham, UK, 2013.

[11] M. Armbrust, S. R. Xin, C. Lian et al., "Spark SQL: relational data processing in Spark," in *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, pp. 1383–1394, ACM, Melbourne, Australia, 2015.

[12] C.-Y. Lin, C.-H. Tsai, C.-P. Lee, and C.-J. Lin, "Supplement materials for 'large-scale logistic regression and linear support vector machines using spark'," in *Proceedings of the IEEE International Conference on Big Data*, 2014.

[13] M. Zaharia, M. Chowdhury, T. Das et al., "Resilient distributed datasets: a fault-tolerant abstraction for in-memory cluster computing," in *Proceedings of the 9th USENIX Symposium on Networked Systems Design and Implementation (NSDI '12)*, San Jose, Calif, USA, April 2012.