Twitter data analysis and visualizations using the R language on top of the Hadoop platform

Martin Sarnovsky, Peter Butka, Andrea Huzvarova
Department of cybernetics and artificial intelligence,
Faculty of electrical engineering and informatics,
Technical university Kosice, Letna 9/A
04001 Kosice, Slovakia
martin.sarnovsky@tuke.sk, peter.butka@tuke.sk, andrea.huzvarova@student.tuke.sk

Abstract— **The main objective of the work presented within this paper was to design and implement the system for twitter data analysis and visualization in R environment using the big data processing technologies. Our focus was to leverage existing big data processing frameworks with its storage and computational capabilities to support the analytical functions implemented in R language. We decided to build the backend on top of the Apache Hadoop framework including the Hadoop HDFS as a distributed filesystem and MapReduce as a distributed computation paradigm. RHadoop packages were then used to connect the R environment to the processing layer and to design and implement the analytical functions in a distributed manner. Visualizations were implemented on top of the solution as a RShiny application.**

I. INTRODUCTION

Nowadays the volume of the data available in different forms is significant and still increasing. The rate of the data increasing is higher than the rate of computational performance. Data processing in such volumes then faces the problem of their processing and storage. Processing and analysis of large volumes of the data also produces new data. On the other hand, processing of large volumes of the data often requires parallel and distributed computation to achieve results in reasonable time, or just to process the amount of the data [1].

Fast and efficient tools are necessary to perform such task applied on big data collections. New models and computing paradigms were designed to support them using hardware resources in form of clusters and other distributed computing architectures. One of the most popular distributed computing paradigms nowadays is a MapReduce [2]. Designed and developed by Google, it is aimed at parallel data processing of the large distributed data collections. The MapReduce processing paradigm is based on two main phases (mapping and reducing), both of them performed in parallel fashion on specified data subsets on multiple computing nodes. Implemented by the Hadoop framework, the distribution logic, load balancing, fault tolerance are main advantages of the solution. Those issues are handled automatically by the framework itself, which enable the developers to be more focused on programming logic. On the other hand, for storage purposes, Hadoop offers HDFS (Hadoop Distributed Filesystem). MapReduce processing has certain limitations and is not very suitable in iterative tasks [3]. Several limitations were removed in the next version of the Hadoop resource management implementation (MapReduce v2) which introduced YARN (Yet another resource negotiator) as the resource manager [5]. This led to a development of more advanced frameworks which remove those limitations, such as in-memory computation frameworks such as Apache Ignite or Apache Spark. Spark also supports cyclic dataflows and in-memory computations which makes it ideal for data processing tool [4].

Various processing and storage tools developed on top of those platforms exists, that extend the Hadoop environment to other areas. HBase, Stratosphere, etc. add the database capabilities to the ecosystem, Hive can be used as a data warehouse and Impala can be used as a querying tool. Also, several machine learning libraries are available. Perhaps a most popular one is Mahout, which contains MapReduce implementations of various machine learning algorithms. Currently Mahout is moving from MapReduce and also support Spark. MLlib is another machine learning library, built on top of the Spark ecosystem, content-wise similar to Mahout [9]. Besides that, several other machine learning tools offer support Hadoop/Spark environments such as H2O and also the more traditional libraries such as Weka or RapidMiner could be user on top of those technologies.

Besides those tools, there are also available tools that enable to connect the popular analytical environments, such as R, to big data processing technologies. For integration of R with Hadoop, RHadoop is available as a set of R packages providing interfaces to HDFS and a set of functions to write MapReduce operations [6, 7]. Other one is Tessera, an open-source computing environment for analysis of large complex data. Tessera enables the data analyst to implement the analysis directly in R and uses Divide&Recombine (D&R) (similar to MapReduce) to run the analysis on the cluster backend. Trelliscope is another tool that can be used in visualization of large scale data analysis [8].

The main goal of this paper is to provide information on the design and implementation of Twitter social network data analysis and visualization tool developed using existing R-based technologies with the use of the private Hadoop cluster. Therefore, our aim was to use R language and RStudio for development on the cloud platform using Cloudera technology. RHadoop was used for the implementation of functions for processing of the large datasets on our cloud infrastructure. The

978-1-5090-5655-2/17/$31.00 ©2017 IEEE

visualization part was implemented using RShiny web-based framework. The main advantage is in the fact that cloud-based solution using RHadoop provides larger storage capacities and data integrity, as well as better computation efficiency, data reliability and failure processing (it is not so critical if some node fails, cluster-based solution can work further). The flexibility of such solution is better, because an addition of new components and extension of technical parameters is much easier. Also, we aimed to integrate the solution into our complex platform providing analytical services in distributed environment [10,11]. We would like to support different types of processes for R-based analytics, from laboratory data analysis processes [12,13], evaluation of learning processes [14,15], to systems for support of teaching process of data analysis methods [16].

The paper is organized as follows. Next chapter describes elements of RHadoop, which is the main part of the architecture on the side of R language codes. In the following chapter, we provide the architecture of the proposed system for analysis of Twitter data. Then, selected dataset and preprocessing steps are described in the next chapter. The following chapter is finally related to the analytical tasks and visualizations designed and implemented within the proposed system. Conclusions and future work ideas are presented at the end of the paper.

II. TECHNOLOGIES USED

RHadoop is a collection of five R packages designed to support the big data processing and analysis tasks in the R environment. Developed by Revolution Analytics, packages are compatible with various distributions of Hadoop frameworks such as Cloudera or Hortonworks, or open source Apache Hadoop distribution. RHadoop consists of five R packages: *rhdfs*, *rmr2*, *rhbase*, *plyrmr* and *ravro* [19].

- rhdfs – provides basic connectivity to a distributed Hadoop filesystem (HDFS). Using the rhdfs package, developers are able to view, read and edit the data stored in HDFS. Rhdfs functions can be divided into 5 subcategories:
  - File manipulation functions – enable developers to access the HDFS and move, copy, remove the data, or change permissions.
  - Read/write functions – enable developers to work with the content of the files
  - Directory functions – dedicated to the creation and modification of the directory tree structure
  - HDFS usage functions – utility functions providing various information about the data in HDFS
  - Initialization functions.
- rmr2 – package providing the set of functions to write a R code that can be transformed into the MapReduce tasks to be deployed in the Hadoop environment
- rhbase – package using to connect to the HBase NOSQL distributed database using Thrift server. Functions contained in this package enables developers to access the data in the HBase tables.
- plyrmr – package that enables to execute data manipulation functions contained in packaged dplyr and reshape2, but on the large sets of data stored in Hadoop clusters. Similarly, to rmr2, it relies on translation of the R code into the MapReduce paradigm.
- ravro – package used to connect to the Avro files from the HDFS.

III. ARCHITECTURE OF THE PROPOSED SYSTEM

In this work we used a small-sized cluster infrastructure that consisted of master node and three worker nodes. The configuration of the Master node was as follows: 64 GB RAM, 8 CPU cores. The worker nodes contained 32 GB RAM and were equipped with 4 CPU cores. Cluster nodes operated the CentOS operating system and Cloudera[1] Hadoop stack (in version 5.6.0.) was used as a Hadoop framework distribution. From available components of the CDH (Cloudera Hadoop) stack we used HDFS (Hadoop Distributed FileSystem, a file storage) and YARN (Yet Another Resource Negotiator, a resource manager). Cluster environment was updated to support the distributed processing of the R functions. For that purposes, the R packages described in the chapter 2 were deployed and configured on each node.

Overall architecture of proposed system is depicted on Figure 1. Hadoop cluster is used for data storage and processing of the analytical functions written in R. Preprocessing and analysis methods are written using the RHadoop packages functions, which enables the code to utilize the cluster framework MapReduce computation paradigm. On top of the R implemented scripts, we have developed a R Shiny application which serves as a user

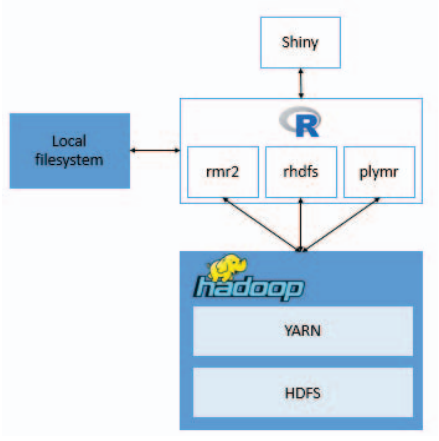

Figure 1. Architecture of the proposed system

interface to the analytical methods provided by the system as well as for visualization purposes.

[1] http://www.cloudera.com

IV. DATASET AND PREPROCESSING

Dataset used in this works is comprised of Twitter social network data. Twitter is a microblogging network for exchange and sharing of the short public messages. Tweets are limited to 140 characters and usually contain hyperlinks, multimedia, hashtags, etc. In some cases, also geolocation is provided, describing location, where the tweet was sent to the network. Tweets data also describes how other users deal with the tweet by retweets and favorite counts, or information about popularity of users in followers count.

In our work we used the Twitter feeds collected and extracted from the data stream platform developed in Urban Sensing[2] project which aimed to develop a platform extracting patterns of use and citizens' perceptions related city spaces, through analysis of user generated content shared by people over social networks. Data are in JSON (Javascript Object Notation) format. The dataset used in this work contained twitter feeds from New Jersey and New York.

Data were streamed into the R environment using the Jsonlite package (*stream_in* function). Then we extracted data only from the New York area using *filter* function in *dplyr* package. Resulting data frame consisted of 679477 records and 30 attributes.

Data cleaning was the next step of the preprocessing phase. Attribute *Datetime_timestamp.date* describing the date and time, when a tweet was created, was divided into the separate attributes representing a year, month, day, hour and minute, *DayOfWeek* function was created to covert the date to a particular weekday.

Among the most important attributes were the ones describing the geolocation of the tweet creation. *Location* and *bearing_from* attributes were transformed into the *Latitude* and *Longitude* attributes. After data cleaning and preprocessing, the data frame contained the attributes:

- Content – the content of a tweet, including hyperlinks, hashtags, etc.
- Text – cleaned content of the feed, only text
- WordCount – wordcount of the text field
- CharCount – characters count in the text field
- ID – twitter user identifier
- Bearing.distance – distance between actual and last recorded user position (at the time of sending the tweet)
- Place.name – city/state name that the user is tweeting in
- Latitude – latitude of the user location
- Longitude – longitude of the user location
- fromLatitude – last recorded user latitude
- fromLongitude – last recorded user longitude
- year – of the tweet creation
- month – of the tweet creation
- day – of the tweet creation
- hour – of the tweet creation
- minute – of the tweet creation
- dayInWeek – particular weekday of the tweet creation

The data were stored in the HDFS and accessed and imported into the R environment using the *rhdfs* package. Preprocessing methods then were implemented using the *plymr* and *rmr2* packages and implemented in a MapReduce-based manner.

V. ANALYSIS AND VISUALIZATIONS

Within the next chapter we provide the details of implemented analyses and their visualizations on the processed New York tweets dataset. We decided to implement four main group of data visualizations. For visualization purposes we utilized several R provided packages, such as *ggmap*, *ggplot*, *leaflet* and *wordcloud*.

A. Analysis of tweets frequencies and their locations in some specific time

This analytical task is based on the computation of the number of tweets produced on some specific place (location) defined by longitude and latitude coordinates. For this reason, we needed to prepare extraction functions of tweets for a specific time (i.e., time frame). Then the MapReduce job was created to effectively count tweets for a set of different coordinates. The input data for this operation was HDFS table of attributes (columns) representing the latitude and longitude of user and counts column. The standard MapReduce idea was then applied to count numbers (sum) of tweets for same coordinates. The result set of the operation was a combination of columns for coordinates and count number for the whole dataset. For our dataset, the minimum count was 1 tweet and maximum number was 839. The computed result set (table) was then used in the visualization part of the application to show counts in RShiny interface on the maps. In this case, users of the application can select a time frame (interval) and the minimum frequency of tweets, the result is the map with places of tweets creation in the selected time interval. *Ggmap* package was used for this visualization. The example of such visualization is shown on Figure 2.

B. Wordcloud

The next analytical task developed within the proposed system was the extraction of typical words from tweets based on the analysis of hashtags (words starting with the

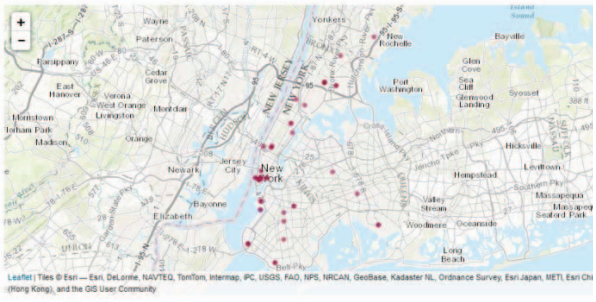

Figure 2. Visualization of tweets frequency and their location

character). After the extraction of hashtags from the dataset, MapReduce job was created for the extraction of

[2] http://urban-sensing.eu/

their numbers. In this case "content" attribute of hashtag was used for map function and their respective counts were acquired using reduce function to get counts for different hashtags. Therefore, the result set of the MapReduce job was the table of words and their counts (frequencies). This result set was then used in visualization part for creation of wordcloud based on the frequencies of hashtags. The example of wordcloud visualization was implemented using *wordcloud* package and is depicted in Figure 3. In RShiny interface the user has the possibility to select the maximum number of words which will be included in the wordcloud, as well as minimum required frequency of the words which can be used for visualization. As we can see, some words are repeated thanks to usage of case-sensitive letters (e.g., "NYC" and "nyc"). In this case, due to fact that case-sensitivity is important in hashtags, we decided to not apply lower-case filter in the preprocessing step.

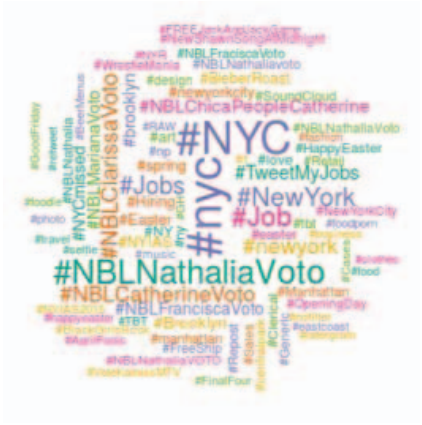

Figure 3. Frequent hashtags visualization

C. *Movement of people*

Another interesting analytical task based on this dataset is to show people in move. It is possible thanks to fact that there are attributes *bearing.from* and *bearing.distance*, which provide last coordinates of the user (if he/she was logged in last 24 hours) and the distance between actual and previous position. In order to process it, we first decomposed *bearing.from* attribute to two columns – *fromLatitude* and *fromLongitude*. Then it was simply possible to provide computation job that filters only the new coordinates without changes of position less than 1000 meters (to setup threshold for map visualizations, i.e., New York area is relatively large, therefore it was not necessary to go for very small distance changes in visualizations).

Pairs of previous and their next coordinates to user were created and inserted into HDFS table. Using this table, we

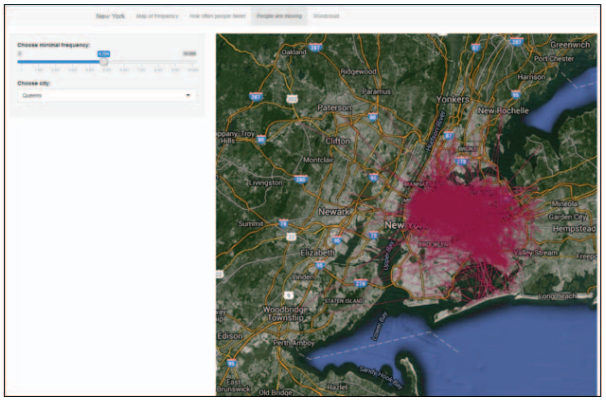

Figure 4. Visualization of persons' movement in the Queens area

are then able to read position changes effectively and work with them or visualize them. The visualization of movement of people is realized in RShiny framework, where user can select one of the New York areas (Queens, Brooklyn, Bronx, Manhattan, Staten Island) and the minimum number of visualized coordinates between 1000 and 10000. If a smaller number of coordinates are selected, it is possible to show particular changes of locations. If we select a larger number of coordinates, it is

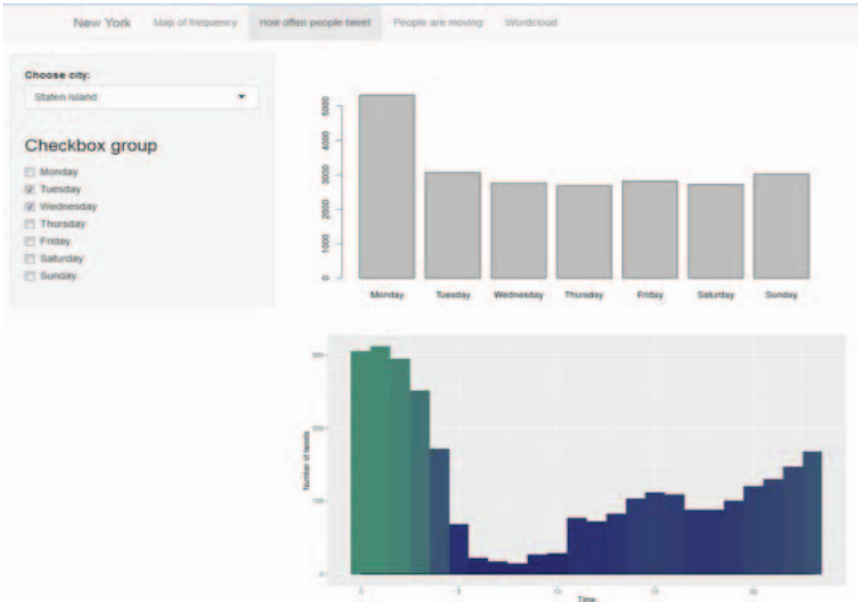

Figure 5. Tweets count based on date, time and location

possible to see better the area where people are moving during the day. The example of such visualization is shown on Figure 4.

D. Tweets according to the location area and daytime

In the last analytical task presented here, we provide a tool for analysis of tweets counts produced for specific date, area and time during the day (hour). Most of the tweets are created during the evening and night hours, while the afternoon and morning hours are not so frequent. The graphs of tweets count within our dataset for days during a week showed that most frequent day for tweets is Monday. Regarding the location or area, we have not found some significant differences between areas (on the larger scale of five large areas already mentioned in previous analytical task). For the implementation of this analytical task, the similar MapReduce approach was applied as is described in subchapter A related to tweets frequency mapping. Again, all the functions related to this analytical task were used within the RShiny web interface, where users can select specific days and area for visualization histograms and counts of tweets for selected options.

VI. Conclusion and Future Work

The main objective of presented paper was to describe the designed and implemented system for twitter data analysis and visualization. It was developed using R and utilized the big data processing technologies. Small-sized Hadoop cluster was deployed and enhanced with RHadoop packages to support the distributed processing of R functions. We developed a set of analytical methods using MapReduce framework from RHadoop package and designed a set of visualizations implemented as Shiny web applications. RHadoop functions were used and utilized in numerous preprocessing, data cleaning and querying methods and proved to be highly-useful and easy to implement for such type of tasks in the selected language and environment.

Acknowledgment

The work presented in this paper was supported by the Slovak Cultural and Educational Grant Agency of Ministry of Education, Science, Research and Sport of the Slovak Republic (KEGA) under grant No. 025TUKE-4/2015 and also by the Slovak Grant Agency of Ministry of Education and Academy of Science of Slovak Republic (VEGA) under grant No. 1/0493/16.

References

[1] J. P. Dijcks, *Big Data for the enterprise*. Oracle White Paper, 2013.

[2] J. Dean, S. Ghemawat, "MapReduce: simplified data processing on large clusters," in Proceedings of OSDI'04: Sixth Symposium on Operating System Design and Implementation, San Francisco, CA, pp. 107-113, 2004.

[3] J. Ishwarappa-Anduranha, "A Brief Introduction on Big Data 5V Characteristics and Hadoop Technology," *Procedia Computer Science*, Odisha, India, pp. 319-324, 2015.

[4] M. Zaharia, M. Chowdhury, M. J. Franklin, S. Shenker, I. Stoica, "Spark: cluster computing with working sets," in Proceedings of the 2nd USENIX conference on Hot topics in cloud computing (HotCloud'10), Berkeley, CA, USA, 2010.

[5] V. K. Vavilapalli et. al., "Apache Hadoop YARN: yet another resource negotiator," in Proceedings of the 4th annual Symposium on Cloud Computing (SOCC '13), ACM, New York, USA, Article 5, 2013.

[6] A. Mittal, S. Pathak, T. Bannard, "RHadoop: An Improved Execution Environment for Restricted MapReduce Programs," Technical paper, 2013.

[7] P. Vignesh, *Big Data Analytics with R and Hadoop*. Birmingham : Packt Publishing Ltd., 2013.

[8] R. Hafen, L. Gosink, J. McDermott, K. Rodland, K. Kleese-Van Dam, W.S. Cleveland, "Trelliscope: A System for Detailed Visualization in the Deep Analysis of Large Complex Data," in Proceedings of the 2013 IEEE Symposium on Large-Scale Data Analysis and Visualization (LDAV), pp. 105-112, 2013.

[9] R. Anil, T. Dunning, E. Friedman, S. Owen, *Mahout in action*. Shelter Island: Manning, 2011.

[10] K. Furdík, J. Paralič, F. Babič, P. Butka, P. Bednár, "Design and Evaluation of a Web System Supporting Various Text Mining Tasks for the Purposes of Education and Research," *Acta Electrotechnica et Informatica*, vol. 10, no. 1, pp. 51-58, 2010.

[11] M. Sarnovsky, P. Butka, P. Bednar, F. Babic, J. Paralič, "Analytical Platform Based on Jbowl Library Providing Text-Mining Services in Distributed Environment," Information and Communication Technology, Third IFIP TC 5/8 International Conference, ICT-EurAsia 2015, and 9th IFIP WG 8.9 Working Conference, CONFENIS 2015, Held as Part of WCC 2015, Daejeon, Korea, October 4–7, *Lecture Notes in Computer Science*, vol. 9357, pp. 310-319, 2015.

[12] V. Gašpar, R. Andoga, "Design of a laboratory information system for data processing and efficiency evaluation," *Acta electrotechnica et informatica*, vol. 15, no. 4, pp. 22-29, 2015.

[13] V. Gašpar, L. Madarász, R. Andoga, "Scientific research information system as a solution for assessing the efficiency of applied research," *Advances in Soft Computing, Intelligent Robotics and Control*, Topics in Intelligent Engineering and Informatics 8, Springer, pp. 273-293, 2014.

[14] F. Babič, J. Paralič, J. Wagner, "Evaluation of user practices during collaborative processes through proposed historical projection," *Acta electrotechnica et informatica*, vol. 10, no. 4, pp. 82-88, 2010.

[15] F. Babič, J. Wagner, P. Bednár, "Java framework for managing semantic repositories based on RDF standard," Acta electrotechnica et informatica, vol. 11, no. 1, pp. 33-37, 2011.

[16] J. Paralič, K. Furdík, M. Paralič, P. Bednár, P. Butka, J. Wagner, "Semantic support for educational IT services," *Acta Electrotechnica et Informatica*, vol. 12, no. 4, pp. 39-46, 2012.