MODERN TECHNOLOGIES FOR DETERMINATION OF PRECIPITATION IN THE RAINIEST COUNTRIES OF THE WORLD

V.V. Golyan, N.V. Golyan, K.R. Galchenko, O.V. Kalinichenko

Abstract: Modern technologies for collection and processing of a great amount of Big Data datasets using the Apache Spark framework are investigated. They provide for modeling of the ecological situation in the rainiest regions of the world, which helps to prevent emergencies and reduces their material and social consequences.

Key words: Big Data, Apache Spark, volume, variety, velocity, value, systems of environmental monitoring, business analysts

INTRODUCTION

Since the problem of conservation the environment is a problem of the international character, one of the main tasks of regional governments is to improve the level of environmental security in the region [1], which causes the need to prevent emergencies and reduce their economic and social consequences.

1. THE PURPOSE AND OBJECTIVES OF THE STUDY

The creation of effective systems for environmental monitoring must provide the solution of two problems - the creation of effective systems of information and intelligent support for decision-making and environmental modeling based on the monitoring data.

Based on the analysis of existing tasks of environmental monitoring, it is shown that to solve the first of these problems we have to adapt a number of existing information technologies to the specific nature of this type of tasks. The main technologies of this type include modern technologies for collecting and processing great amounts of Big Data datasets using the Apache Spark framework [2-3].

2. ANALYSIS OF THE SUBJECT DOMAIN

The term "Big Data" was introduced by Clifford Lynch, the editor of the journal "Nature", who analyzed the phenomenon of big data and their significance for science in the same journal in 2008. He collected material on the phenomenon of data explosion with regard to the volume and diversity, as well as technological prospects in the paradigm of a probable transition from "quantity to quality." The concept of big data implies work with the information of a very great amount and diverse composition, which is often renewable and can be found in different sources for increasing work efficiency, creating new products, and increasing competitiveness.

When defining the Big Data concept, one uses four V: Volume, Variety, Velocity, and Value, namely the definition sounds like this: Big Data is "a new generation of technologies and architectures for the economic gaining of value from a great amount of heteromorphic data through their rapid capture, processing and analysis."

Working with big data differs from the usual business intelligence process, where simple addition of the known values gives the result: for example, the total of adding settled account data becomes the sales volume throughout the year. When working with big data, we obtain the result in the process of cleaning them by sequential modeling: first a hypothesis is put forward, a statistical, visual or semantic model is built, on its basis the correctness of the hypothesis is checked, and only then the following hypothesis is put forward. This process requires from the researcher either to interpret the visual values, or to compose interactive queries based on knowledge or development of adaptive algorithms of "computer learning" capable of obtaining the desired result.

Consequently, Big Data in information technology is a series of approaches, tools and methods for processing structured and unstructured data of huge volumes and significant diversity for obtaining results that are perceived by a human being. Big Data are effective in the conditions of continuous growth, distribution in numerous nodes of the computer network that are alternative to traditional database

management systems and solutions of the Business Intelligence class. This series includes massively parallel processing capabilities of indefinitely structured data, primarily by NOSQL solutions using Mapreduce algorithms, software frameworks and Hadoop project libraries.

Installing add-ons and configuring them to work with great amounts of data on a PC takes a long time. Using IBM Bluemix, in order to start working from Big Data, you have to start the Apache Spark service.

Apache Spark is an open-source programming framework for implementing distributed processing of unstructured and poorly structured data that is part of the ecosystem of Hadoop projects.

The Spark extension allows you to use the Sql query to work with data.

To load data, you have to create a repository and then add the file in the CVS format.

In the paper the pandas library is used, which is the Python software library for data processing and analysis. The pandas work with data is built on top of the Numpy library, which is a low-level tool. The pandas provides special data structures and operations for manipulating numeric tables and time series.

3. DESCRIPTION OF THE ADOPTED DESIGN DECISIONS

The data downloaded from the undata site are used in the paper. These are annual precipitation data from different countries presented in the form of statistical databases and provided by the United Nations Statistics Division. The precipitation measurements are presented in millions of cubic meters. In order to get a set of data, one has to download the file:

https://cdsax.cloudant.com/public-samples/test/precipitation.csv.

First, we connect the necessary libraries. The following line in the field of the workspace is entered:

import requests, STRINGIO, pandas as pd, json, re After that, press ► (Run Cell).

The following function allows to quickly access the data:

```
def get_file_content(credentials):
    """For given credentials, this functions returns a StringIO object containing the file content."""
    url1 = ''.join([credentials['auth_url'], '/v3/auth/tokens'])
    data = {'auth': {'identity': {'methods': ['password'],
     'password': {'user': {'name': credentials ['username'],'domain': {'id': credentials['domain_id']},
     'password': credentials ['password']}}}}}
    headers1 = {'Content-Type': 'application/json'}
    resp1 = requests.post(url=url1, data=json.dumps(data), headers= headers1)
    resp1_body = resp1.json()
    for e1 in resp1_body['token']['catalog']:
     if(e1['type']=='object-store'):
    for e2 in e1['endpoints']:
     if(e2['interface']=='public'and e2['region']==credentials['region']):
          url2 = ''.join([e2 ['url'],'/', credentials['container'], '/', credentials['filename']])
    s_subject_token = resp1. headers['x-subject-token']
    headers2 = {'X-Auth-Token': s_ subject_token, 'accept': 'application/ json'}
    resp2 = requests.get(url=url2, headers=headers2)
    return StringIO.StringIO(resp2. content)
```

If you click in the next field of the workspace and press "Insert to code" under the added file on the right, the following code is added to the empty field:

```
credentials_1 = {
'auth_url':'https://identity.open. softlayer.com', 'project':'object_ storage_88cdff6d_f018_4d15_ b7d2_743c327661e7', 'project_id':'617 248f36c884ad48c48821913f5cc9a',
'region':'dallas',
'user_id':'b8d5f8b21be14c1699ede267 a2d9c086', 'domain_id':'070ae40d50cc46 23a24b58e1e9c0bee0',
'domain_name':'1028921', 'usernam e':'Admin_39d9000db67f5cdf8249168e11 05bb888d159835', 'password':"""N!s0. GMu9gz7{ce&""",
'filename':'precipitation (1).csv',
```

```
'container':'notebooks',
'tenantId':'s689-12c938f85f862e-d53ac596a700'
}
```

These credentials are needed to load the data file into the pandas.DataFrame. To load the data into the pandas.DataFrame, run the following code in a new field:

```
content_string = get_file_content(credentials_1)
precipitation_df = pd.read_csv(content_string)
```

Now the data is in the memory, so you can investigate and manipulate them. First, let us display the first 5 results. To do this, execute the following command:

```
precipitation_df.head()
```

The answer will be displayed immediately after the execution below the field in which the command has been executed. It looks like this:

Out[5]:

	Country or Area	1990	1995	1996	1997	1998	1999	2000	2001	200
0	Albania	28385.000000	40311.000000	0.000000	0.000000	0.0	38284.000000	30683.000000	30491.000000	356
1	Algeria	76160.000000	90270.000000	53380.000000	74460.000000	66470.0	50150.000000	64430.000000	43840.000000	375
2	Andorra	539.947998	510.673004	560.340027	434.475006	254.0	450.151001	518.666016	456.626007	562
3	Anguilla	93.099998	100.730003	0.000000	0.000000	0.0	0.000000	68.190002	70.730003	68.
4	Antigua and Barbuda	300.299988	374.500000	323.299988	279.200012	384.5	426.799988	249.600006	238.000000	266

As can be seen, each column in the table contains: the name of the country or region where the measurements were taken; the annual rainfall for the period from 1995 to 2009.

Using the Dataframe API, one can display all the countries or regions whose precipitation measurements are presented in the table. To display these countries, run the following code in a new field:

```
precipitation_df['Country or Area'].values
```

Out[6]:
```
array(['Albania', 'Algeria', 'Andorra', 'Anguilla', 'Antigua and Barbuda',
       'Armenia', 'Azerbaijan', 'Bahrain', 'Barbados', 'Belarus',
       'Belgium', 'Belize', 'Benin', 'Bermuda', 'Bosnia and Herzegovina',
       'Botswana', 'British Virgin Islands', 'Brunei Darussalam',
       'Cameroon', 'Central African Republic', 'Chile', 'China',
       'China, Hong Kong SAR', 'China, Macao SAR', 'Colombia',
       "Cote d'Ivoire", 'Croatia', 'Cuba', 'Cyprus', 'Czech Republic',
       'Denmark', 'Dominican Republic', 'Ecuador', 'Egypt', 'Estonia',
       'Finland', 'France', 'Gambia', 'Georgia', 'Germany', 'Guinea',
       'Hungary', 'India', 'Iraq', 'Israel', 'Italy', 'Jamaica', 'Jordan',
       'Kazakhstan', 'Kuwait', 'Kyrgyzstan', 'Latvia', 'Lebanon',
       'Lithuania', 'Luxembourg', 'Madagascar', 'Maldives', 'Malta',
       'Marshall Islands', 'Mauritius', 'Monaco', 'Morocco', 'Netherlands',
       'Oman', 'Panama', 'Paraguay', 'Poland', 'Portugal', 'Qatar',
       'Republic of Moldova', 'Romania', 'Senegal', 'Serbia', 'Singapore',
       'Slovakia', 'Slovenia', 'South Africa', 'Spain', 'Sri Lanka',
       'Sweden', 'Switzerland', 'Syrian Arab Republic',
       'The Former Yugoslav Rep. of  Macedonia', 'Togo',
       'Trinidad and Tobago', 'Tunisia', 'Turkey', 'United Kingdom',
       'Venezuela', 'Yemen', 'Zimbabwe'], dtype=object)
```

After obtaining the tables, we proceed to the graphical representation of the data. To do this, the package for creating Matplotlib graphs is used.

We present the results in the form of built-in diagrams. To do this, run the following code in a new field:

```
%matplotlib inline
```

Let us display a graph of the amount of precipitation in the countries with the heaviest rainfall. At the end of the table we add a column that contains the amount of precipitation for the whole period of time:

```
precipitation_df["SUM"] = precipitation_df.sum(axis=1)
```

Make sure that the column is added:

```
precipitation_df.head()
```

in the same period of time

Out[13]:

2	2003	2004	2005	2006	2007	2008	2009	SUM
83.000000	27893.000000	42787.000000	42840.000000	32380.000000	30964.000000	0.000000	0.000000	380901.000000
17.000000	0.000000	0.000000	0.000000	0.000000	0.000000	100000.000000	0.000000	656477.000000
.559021	566.583008	567.044006	530.278015	353.220001	306.630005	0.000000	0.000000	6614.193115
190002	108.769997	84.250000	124.400002	99.550003	86.290001	96.889999	71.080002	1072.170013
600006	253.899994	426.899994	371.000000	332.799988	293.600006	392.500000	276.899994	5192.399963

Fig 3.1-The table after adding the precipitation amount column

Next, let us sort the Dataframe, taking into account the total amount of precipitation, and display the countries or regions with the heaviest total precipitation:

```
precipitation_sorted_df = precipitation_df.sort_values(by=”SUM”,ascending=False) top5_sums = pd.Series(precipitation_sorted_df[“SUM”].head(5))
top5_sums
```

```
Out[14]: Country or Area
         China        59269500.0000
         Colombia     35600950.0000
         Venezuela    22160738.6250
         India        16000000.0000
         Chile        15940757.0625
         Name: SUM, dtype: float64
```

Fig 3.2 - The countries with the heaviest total precipitation

Now we present five countries with the heaviest total precipitation in the form of a linear graph for a more convenient analysis of the results:

```
top5_bars = precipitation_sorted_df[years][0:5].transpose()
ax = top5_bars.plot(figsize=(10,8), marker='o', linestyle='-', title="Top 5 Countries with highest Precipitation")
ax.set_xlabel("Years")
ax.set_ylabel("Precipitation (million cubic meters)");
```

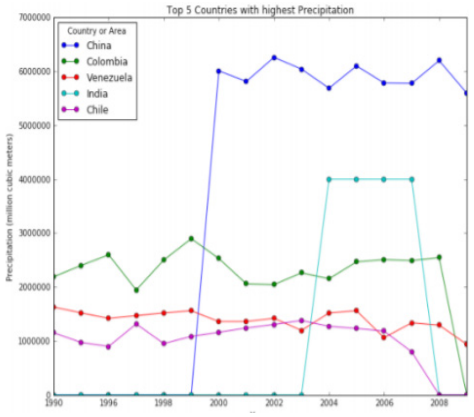

Fig 3.3 - The precipitation graph in the 5 rainiest countries of the world

The following code, run in an empty field, creates a circular diagram of the annual precipitation distribution in the five rainiest regions:

```
precipitation_sums = top5_sums
other_sums = precipitation_sorted_df["SUM"][5:].sum()
precipitation_sums["Other"] = other_sums

plt.axis('equal')
plt.title("Annual precipitation percentage",y=1.08)
plt.pie(
precipitation_sums,
labels=precipitation_sums.index,
colors=['blue', 'green', 'red', 'turquoise', 'magenta','yellow'],
autopct="%1.2f%%",
radius=1.25);
```

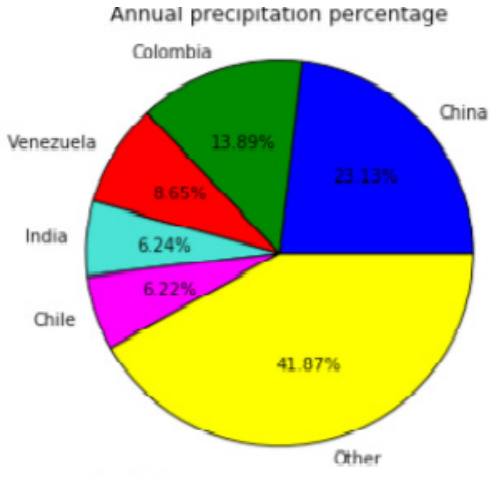

Fig 3.4 - The precipitation graph in the 5 rainiest countries

If we compare the total amount of precipitation for the five leading countries, we will be able to see that China has the greatest value of the annual precipitation followed by Colombia. In the linear graph it is convenient to compare the results of measurements by months; to compare the annual values it is more convenient to use a pie graph.

The pie graph shows that almost a quarter of all the recorded precipitation fell in China, and more than half of the precipitation fell in five countries with the heaviest total rainfall.

CONCLUSIONS

Modern technologies for collecting and processing great amounts of the Big Data datasets using the Apache Spark framework allows the modeling of the ecological situation in five rainiest regions of the world.

LIST OF REFERENCES.

[1] **Morozhenko V.** Stan navkolozemnogo prostory. Ekologij Zemli, ii zv'jazok z problemamu ozonosferu i zminu klimaty // Visnuk NKA Ykrainu. – 2001. -№1. – s. 50-59.

[2] Big Data Visualization: Turning Big Data into Big Insights. The Rise of Visualization-based Data Discovery Tools. White Paper. Intel IT Center. March 2013.

[3] http://www.intel.com/content/dam/www/public/us/en/documents/white-papers/big-data

8. Information about the Authors:

V.V. Golyan
Kharkov National University of Radioelectronics, Department of Software Engineering, 61166, Kharkov, Lenin Avenue 9
e-mail: vira.golan@nure.ua

N.V. Golyan
Kharkov National University of Radioelectronics, Department of Software Engineering, 61166, Kharkov, Lenin Avenue 9,
e-mail: nata2012.nn@gmail.com

K.R. Galchenko
Kharkov National University of Radioelectronics, 61166, Kharkov, Lenin Avenue 9,
e-mail: nata2012.nn@gmail.com

O.V. Kalinichenko
Kharkov National University of Radioelectronics, Department of Software Engineering, 61166, Kharkov, Lenin Avenue 9
e-mail: olga.kalynychenko@nure.ua