Big Data Analysis on Storm Events

in the U.S.

Cloud Computing and Big Data



Faisal Khan

TABLE OF CONTENTS

[1. Executive Summary 3](#_Toc481432164)

[2. Project Apparatus 4](#_Toc481432165)

[3. Dataset 4](#_Toc481432166)

[3.1 Dataset Source 4](#_Toc481432167)

[3.2 Dataset Content 5](#_Toc481432168)

[3.3 Preliminary Analysis of Dataset 6](#_Toc481432169)

[4. Map and Reduce Examples 7](#_Toc481432170)

[4.1 Introduction of Hadoop Streaming 7](#_Toc481432171)

[4.2 Map and Reduce Code 7](#_Toc481432172)

[4.3 Testing Code 8](#_Toc481432173)

[4.4 Running Code 9](#_Toc481432174)

[5. Map and Reduce Result 10](#_Toc481432175)

[6. Problems 13](#_Toc481432176)

[7. Further Study 13](#_Toc481432177)

[8. Conclusion 13](#_Toc481432178)

[9. Reference 14](#_Toc481432179)

1. **Executive Summary**

This project focused on using Hadoop to do MapReduce jobs. Project dataset was about the storm events happened in the U.S. from 2012 to 2016. Hadoop was installed in the Ubuntu Linux system. Datasets were downloaded from the U.S. Government’s open data (<https://www.data.gov/>) and stored in the Hadoop DHFS. MapReduce jobs were coded in Python language. Hadoop Streaming was used to run the MapReduce jobs in Python. The result of the MapReduce jobs was converted into Excel2016 format. Graphs and Charts were plotted using Excel.

1. **Project Apparatus**

Table 2.1 Software Table

|  |  |  |
| --- | --- | --- |
| **Apparatus** | **Version/Model Number** | **Quantity** |
| VMware Workstation Pro | 12.5 | 1 |
| Ubuntu Linux | 16.4 | 1 |
| Hadoop | 1.2.1 | 1 |
| Python | 2.7 | 1 |
| Vim | 7.4.1829 | 1 |
| Microsoft Excel | 2016 | 1 |

1. **Dataset**
   1. Dataset Source

Dataset were downloaded from the DATA.GOV website. This website is the home of the U.S. government’s open data. This project dataset was from the NCDC Strom Events Database. NOAA’s National Weather Service entered this database. Bulk data were available in comma-separated files(CSV). This database contained storm events details from 1950 to 2017. In this project, only the datasets from 2012 to 2015 were analyzed. The downloaded files were compressed files (.gz file) which needed to be extracted first. Open the file in Excel to see the dataset content.



Figure 3.1 Dataset File

* 1. Dataset Content

Table 3.1 Dataset Attributes and Example

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Content** | **Example** |
| EPISODE\_ID | Assigned by NWS to denote the storm episode | 108769 |
| EVENT\_ID | The primary database key field | 651823 |
| STATE | The state name where the event occurred | SOUTH CAROLINA |
| STATE\_FIPS | A unique number assigned to the county by the National Institute for Standards and Technology | 45 |
| YEAR | Four digital year for the event in this record | 2016 |
| MONTH\_NAME | Name of the month for the event in this record | July |
| EVENT\_TYPE | Most accurately describes the meteorological event | Heavy Rain |
| CZ\_TYPE | Indicates whether the event happened in a (C) county/parish, (Z) zone or (M) marine | C |
| CZ\_FIPS | A unique number assigned to the county by the National Institute for Standards and Technology (NIST) or NWS Forecast Zone Number | 91 |
| CZ\_NAME | County/Parish, Zone or Marine Name assigned to the county FIPS number or NWS Forecast Zone | YORK |
| WFO | National Weather Service Forecast Office’s area of responsibility (County Warning Area) in which the event occurred | GSP |
| BEGIN\_DATE\_TIME | The begin time of the event, MM/DD/YYYY 24-hour time | 7/15/2016 17:15:00 |
| CZ\_TIMEZONE | Time Zone for the County/Parish, Zone, or Marine Name) | EST-5 |
| END\_DATE\_TIME | The end time of the event, MM/DD/YYYY 24-hour time | 7/15/2016 17:25:00 |

* 1. Preliminary Analysis of Dataset

From the storm events dataset, we can gather the information below:

1. Which state had the most storm events?

To get the answer for this question, we need to count the total events happened in each state.

1. How many events happened in each month in a state?

To solve this problem, we need to calculate the storm events happened in each state each month. The result of each state should be in one line, each line had 12 numbers represent the event times of each month.

1. Which type of events happened most?

The storm events had different types including hail, thunderstorm wind, flood, winter storm, winter weather, etc. To get the answer for this question, we need to calculate the times each type event happened.

1. What was the total duration time of the storm events for a year in a state?

The dataset has the begin time and end time of events, to get the duration time, we need to subtract the begin time form the end time, add all the duration time together for each state.

1. **Map and Reduce Examples**
   1. Introduction of Hadoop Streaming

Hadoop streaming is a utility that allows you to create and run Map/Reduce jobs with any executable or script as the mapper and/or the reducer. Languages like C, C++, Python, and Shell can all be run in Hadoop streaming. Any job in Hadoop must have two phases: mapper and reducer. Codes were written for the mapper and the reducer in python script to run it under Hadoop.

* 1. Map and Reduce Code

1. Mapper Code Example

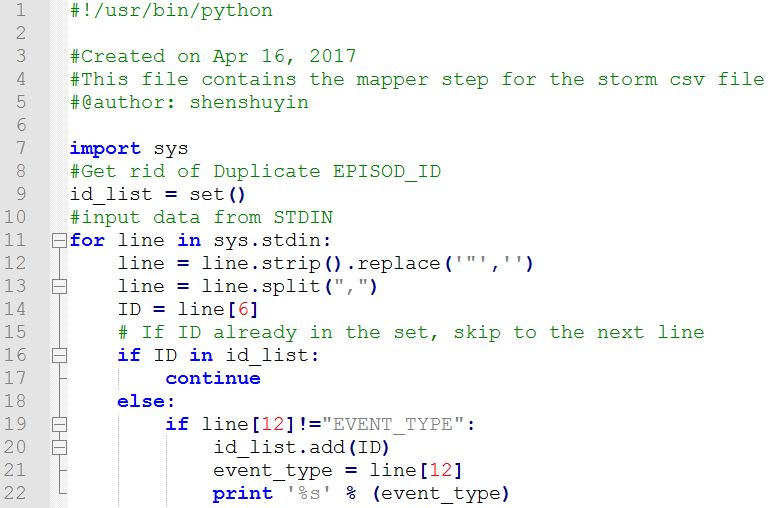


Figure 4.1 Mapper Code Example

This mapper program can read data from the standard input line by line, split the data into list. Different attribute can be located with the index. The print function can send the content to the standard output. The standard output can send the content to the reducer.

1. Reducer Code Example

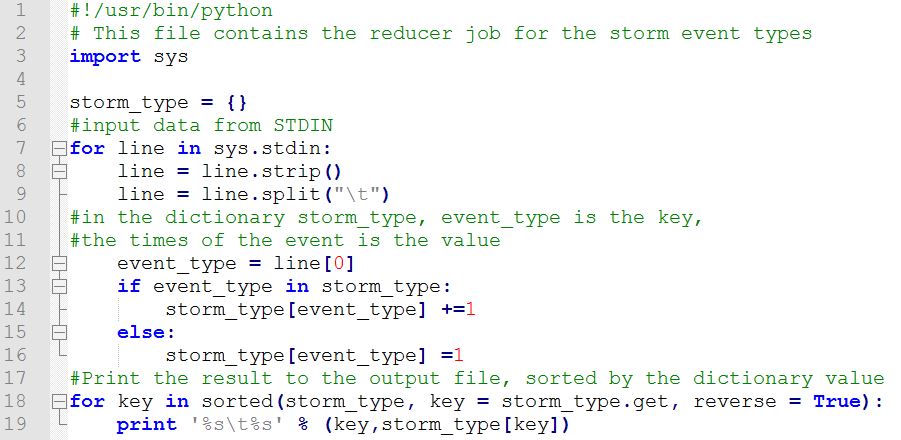


Figure 4.2 Reducer Code Example

This reducer program can read data line by line from the standard input, which was sent by the mapper program. This program splits the line and stored the data into a dictionary. The key of the dictionary is the event type, and the value of the dictionary is the count of the key event.

* 1. Testing Code

Before run the Hadoop MapReduce, we need to test the Python code. Firstly, make sure this file has execution permission using the command below:

$ chmod +x <file location dictionary>/mapper.py

$ chmod +x <file location dictionary>/reducer.py

Secondly, test the code using the command below:

$ cat <test file location directory>/<test dataset file name> | <mapper file location directory>/mapper.py | sort | <reducer file location directory>/reducer.py

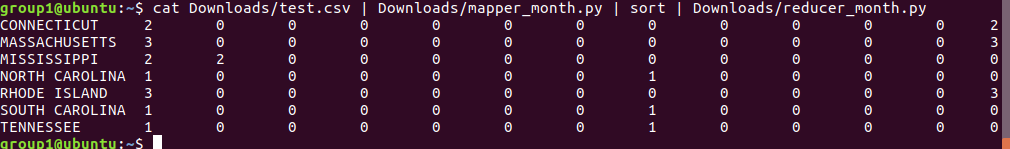


Figure 4.3 Python Code Test Example

* 1. Running Code

To run the Hadoop MapReduce job, we need to locate the hadoop-streaming.jar. In this project, the hadoop-streaming file was in /usr/local/Hadoop/contrib/streaming. Then run the command below:

$ hadoop jar /usr/local/Hadoop/contrib/streaming/hadoop-streaming-1.2.1.jar \

> -mapper mapper.py

> -reducer reducer.py

> -input /<input dataset location dictionary in the DHFS> \

> -output /<output dictionary in the DHFS> \

> -file <mapper file location dictionary>/mapper.py \

> -file <reducer file location dictionary>/reducer.py \

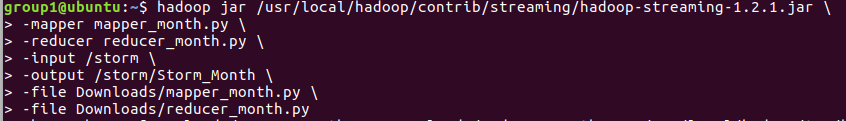


Figure 4.4 Hadoop Streaming MapReduce Command Example

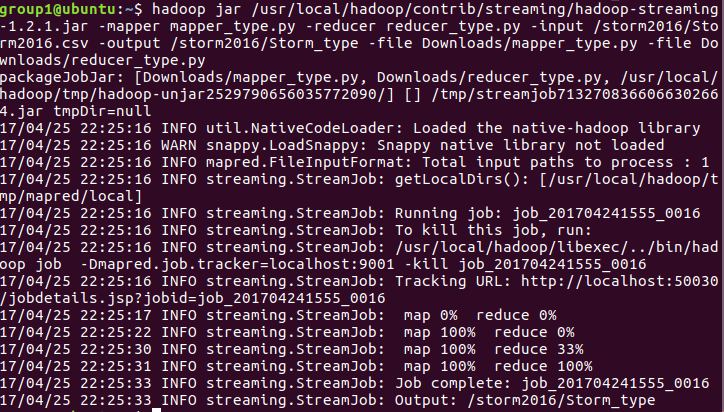


Figure 4.5 MapReduce Job Running Example

1. Map and Reduce Result
2. Storm Events Month Analysis Jobs Output Example

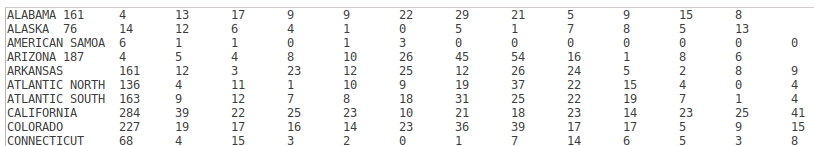


Figure 5.1a Storm events total for each state in 2016(JAN-DEC)

The output file located in the DHFS name “part-00000”. Copy this content into the Excel file and draw graphs.

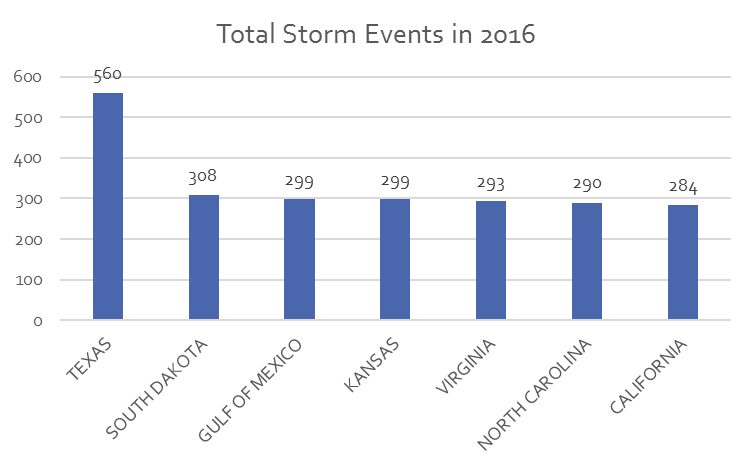


Figure 5.1b Top 7 States for the most storm occurrences in 2016

Figure 5.1c Storm events comparison (2015 vs 2016)

Figure 5.1d Storm events comparison in CT (2015 vs 2016)

1. Storm Events Type Analysis Jobs Output Example

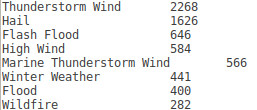


Figure 5.2a Storm event types in 2016

Figure 5.2b Storm event types comparison (2012-2016)

Figure 5.2c Winter storm events types comparison (2012-2016)

1. Storm Events Duration Analysis Jobs Output Example

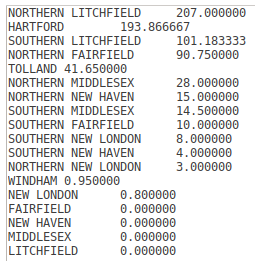


Figure 5.3a Storm duration in Connecticut in 2016

Figure 5.3b Storm duration in CT 2016

1. **Problems**

When we were testing the duration job, we got output for ‘400 Days’, which was way over a year. Then we rechecked the datasets and found that, every row had an EPISODE\_ID to identify a storm episode, and EVENT\_ID to identify an event. There were records having the same EPISODE\_ID, but different EVENT\_ID.

We then filtered the dataset in the mapper job and got rid of the duplicate EPISODE\_IDs.

1. **Further Study**

For the further study, we need to analyze more year data and try to make some prediction using the neural network, machine learning and deep learning. In the further study, we want to solve the problem below:

* What type of climate changes are occurring?
* Is Global Warming affecting our weather?
* Is the world getting WARMER?!!

1. **Conclusion**

In Hadoop, there is a Java program called Hadoop streaming-jar. This program internally reads (stdin) and prints out (stdout) line by line. Therefore, Python or other coding language can read each line as a string and parse it by using functions like strip and split (",").

Before running the codes, we should get to know of the dataset. We need to know the attribute and details about the data. Know what we can get from the data before we do the MapReduce Jobs or think about analyzing the data.

1. **Reference**
2. HADOOP-STREAMING. (n.d.). Retrieved April 22, 2017, from

<https://www.tutorialspoint.com/hadoop/hadoop_streaming.htm>

1. Hadoop Streaming. (Aug 4th, 2013). Retrieved April 22, 2017, from

<https://hadoop.apache.org/docs/r1.2.1/streaming.html>

1. Rathbone, Matthew. Hadoop Python MapReduce Tutorial for Beginners. (Nov 17th, 2013). Retrieved April 16, 2017, from

<https://blog.matthewrathbone.com/2013/11/17/python-map-reduce-on-hadoop-a-beginners-tutorial.html>

1. Map Reduce example: Take Average. (n.d.). Retrieved April 16, 2017, from

<http://rare-chiller-615.appspot.com/mr1.html>

1. Noll, Michael G. Writing an Hadoop MapReduce Program in Python. (n.d.). Retrieved April 16, 2017, from

<http://www.michael-noll.com/tutorials/writing-an-hadoop-mapreduce-program-in-python/>