Analysis of Crime Data and Finding Frequent Patterns using Hadoop and Data Analytic Techniques

Aniket B. Wakde
Pimpri Chinchwad College of Engineering
Pune
411044

Shravani J. Uttarwar
Pimpri Chinchwad College of Engineering,
Pune
411044

Sudarshan D. Waydande
Pimpri Chinchwad College of Engineering
Pune
411044

Purvesh Shende
Pimpri Chinchwad College of Engineering
Pune
411044

Ganesh Deshmukh
Pimpri Chinchwad College of Engineering
Pune
411044

ABSTRACT
With continually increasing population and violations, rate of crime is also increasing. Analyzing such rapidly increasing data regularly is a huge issue for police department. This is extremely important to guard the residents of the nation from violations. Certain patterns must be discovered, examined and talked about to take well informed decisions so that law and orders can be kept up legitimately. Hadoop is one of an open source programming framework that supports the processing and storage of extremely large data sets in a distributed computing environment. Pig Latin is the scripting language to construct MapReduce programs for an Apache project which runs on Hadoop. The benefit of using Pig Latin is that fewer lines of code has to be written which reduces overall development and testing time. Therefore, pig technology works more efficiently than MapReduce which is presented in this paper using case study of Crime Data Analysis.

General Terms
Hadoop ecosystem, Pattern recognition, Public Security etc.

Keywords
Big Data, Hadoop, Map-Reduce, Pig etc.

1. INTRODUCTION
Big data is a collection of voluminous amount of informational data. Big data philosophy encompasses unstructured, semi-structured and structured data, however the main focus is on unstructured data. Big data "size" is a constantly moving target, as of 2018 data is ranging from some Petabyte to Exabyte.

Big data can be described by following characteristics:

Volume: The quantity of generated and stored data. The size of the data determines the value and potential insight, and whether it can be considered big data or not.

Variety: The type and nature of the data. This helps people who analyse it to effectively use the resulting insight. Big data draws from text, images, audio, video; plus it completes missing pieces through data fusion.

Velocity: In this context, the speed at which the data is generated and processed to meet the demands and challenges that lie in the path of growth and development. Big data is often available in real-time.

Value: This characteristic refers to the intrinsic value contained in big data.

Veracity The data quality of captured data can vary greatly, affecting the accurate analysis.

Hadoop is an open source, java based programming framework that covers all the aspects of big data like volume, Variety, Velocity, Value and Veracity.

HDFS is used to store voluminous amount of structured, Semi-structured or unstructured data. HDFS acts as a storage system for Hadoop distributed framework. Hadoop basically fallows master and slave architecture with namenode as master and other processing datanodes as slave systems. Namenode contains metadata of slave systems and actual processing is done on datanodes.

MapReduce is a programming paradigm in Hadoop framework which allows user to write code in java for the anlysis of bigdata. It works in two phases Map and Reduce. In first phase it takes set of data and convert it into another set of data where individual elements broken down into key value pair. This set of tuples are considered as an input for second phase. In this phase data tuples get combines into smaller data tuples. In Mapreduce we can easily scale data processing over multiple computing nodes.

With the introduction of YARN (Yet Another Resource Negotiator) in later releases of Hadoop, various interesting and wonderful components get integrated with Hadoop. With the help of this components one can store, process and analyse the data lot more efficiently, thus aiding for exploration of data for undiscovered facts at a smooth pace. Some of these components that works on top of Hadoop are Pig, Hive, Hbase, Squoop, Flume etc.

2. COMPARATIVE ANALYSIS OF PIG AND MAP -REDUCE
Pig technology can handle almost all operations which Map-Reduce can handle. But Pig have some advantages over the Map-Reduce. Pig is a high level data flow language which is easy to understand and learn rather than low level processing paradigm like Map-Reduce. Pig is a scripting language, so no need to write complex program. Pig technology reduces overall development time as well as testing time. Pig provides built in support for data operation like join, filter, ordering, sorting etc. Pig provides nested data types like tuple, bag, map etc.

Table 1. Comparison: Map-Reduce Vs Pig

Factor	Map-Reduce	Pig
Language Type	Compiled type language	Scripting language
Level of abstraction	Low level of abstraction	High level of abstraction
Lines of code	More lines of Code required	Less lines of code required
Development	More Development is required	Less Development is required
Code Efficiency	Code efficiency is more	Code efficiency is less

3. PROPOSED FRAMEWORK

3.1 Loading:

If we want to process any kind of data for anlysis purpose then first step is to load data in Hadoop environment i.e. in HDFS. There are number of components available in market which works on top of hadoop which can be used to load the data. **Apache Sqoop** is used to move voluminous amount of structured data in hdfs. Sqoop has a generic JDBC connector which helps to move data from all kinds of RDBMS's along with all famous SQL services such as MySQL and oracle's SQL+. For storing, Aggregating and moving unstructured data **Apache Flume** is used which is distributed and very reliable service.

3.2 Storing and Processing:

The Hadoop Distributed File System is the primary storage system used by Hadoop applications. HDFS is a distributed file system and a framework provided by Hadoop for the analysis and transformation of huge dataset which uses MapReduce paradigm. It Provides high performance access to data across Hadoop cluster, HDFS has become a key tool for managing pools of big data and supporting big data analytics applications.

The storage, access and modification jobs are done by two different tasks, the Job tracker (master) and the Task tracker (slave). Master job Tracker manages the resources and tracks the consumption and availability of resources. It also schedules the job component tasks on the slaves and monitor it. While Slave task tracker execute the tasks as per the directions of the master and provide task-status information to the master periodically.

The intercommunication among tasks and nodes is done via periodic messages called heartbeats. One of the ways of achieving this is MapReduce, It is a software framework which is used to write applications easily to process huge amount of data simultaneously on large cluster in reliable and fault tolerant manner. It takes input and gives the output in form of key-value pairs.

The Map phase processes each record sequentially and independently on every node and generates intermediate key-value pairs.

Map(k1, v1) → list(k2, v2)

The Reduce phase takes the output of the Map phase. It processes and merges all the intermediate values to give the final output, again in form of key-value pairs.

Reduce(k2, list (v2)) → list(k3, v3)

The output gets sorted after each phase, thus providing the user with the aggregated output from all nodes in an orderly fashion.

3.3 Analysis:

After collecting and storing data into HDFS next step is to analyses that data. Whatever the data is collected, it has to be processed to extract some meaningful and useful information which supports for decision making. Therefore the analysts need to come up with a good technique for the same. However, Writing MapReduce requires basic knowledge of Java along with sound programming skills. Even after writing the code, which is in itself a labor intensive task, with additional time is required for the review of code and its quality assessment. But now, analysts have additional options of using the Pig Latin. Pig Latin is the scripting language to construct MapReduce programs for an Apache project which runs on Hadoop. The benefit of using this is that fewer lines of code has to be written which reduces overall development and testing time. These scripts take just about 5% time compared to writing MR programs but 50% more time in execution. Although Pig scripts are 50% slower in execution compared to MR programs, they still are very effective in increasing productivity of data engineers and analysts by saving lots of time in writing phase.

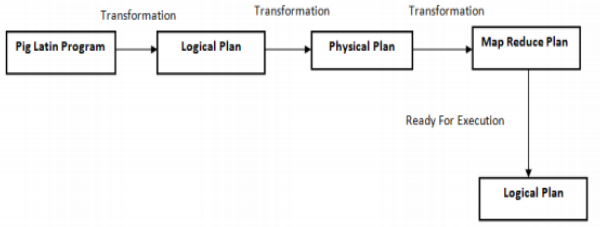

Fig 1: Pig Latin overflow

Apache Pig is a high level platform for creating programs that run on apache Hadoop. Pig can execute its Hadoop jobs in mapReduce, apche Tez or Apche Spark. Pig latin abstracts the programming from the java mapreduce idiom into a notation which makes MapReduce programming high level, similar to that of SQL for relational database management systems.

4. RESEARCH QUESTIONS

Some of the problem statements along which the analysis has been done in this paper:

1. **Total Number of crimes from the year 2012-2017:**

With the ever increasing population and crime rates, certain trends must be discovered, so that law and order can be maintained properly and there is a sense of safety and well-being among the citizens of the country.

2. **Total Number of crimes in 2016 with IUCR codes:**

If the number of complaints from a particular state is

found to be very high, extra security must be provided to the residents there by increasing police presence, quick redressal of complaints and strict vigilance.

3. **Total Number of Crimes in 2012 by Type:**
It includes crime in which the objective is violent for example murder or in which violence means to an end for example robbery.

4. **Total Number of crimes in 2012 by Description:**
Crimes against women are becoming an increasingly worrying and disturbing problem for the government. The number of such crimes must be found, especially the ones against young women (age between 18-30 years)

Hadoop Cluster Mode: Fully-Distributed Hadoop cluster mode is used to analyze the crime data.

Data Set: The data was in .csv Format, that is each line represents data record and each record has one or more field separated by commas.

Data Set Description: The data set includes the following fields:

- ID: chararray
- Case_Number: chararray
- Date: chararray
- Block: chararray
- IUCR: chararray
- Primary_Type: chararray
- Description: chararray
- Location_Description: chararray
- Arrest: chararray
- Domestic: chararray
- Beat: chararray
- District: chararray
- Ward: chararray
- Community_Area: chararray
- FBI_Code: chararray
- X_Coordinate: chararray
- Y_Coordinate: chararray
- Year: chararray
- Updated_On: chararray
- Latitude: chararray
- Longitude: chararray
- Location: chararray

Tools and Technologies used:

1. Hadoop
2. Pig

5. EXPERIMENTAL FINDINGS:

5.1 Total Number of crimes from the year 2012-2017

Input: Crime data set

Output: Total number of crime from the year 2012-2017

Algorithm used in Grunt Shell:

1. Enter into the Grunt shell using command: Pig
2. X = Load the data set using Pig Storage;
3. Y = For each X generate the crime by year;
4. Z = Group by year;
5. Data = For each Z generate group, SUM(X.year);
6. Store output.

5.2 Total number of crimes in 2016 with IUCR code

Input: Crime data set

Output: Total number of crimes with IUCR code

Algorithm used in Grunt Shell:

1. Enter into the Grunt shell using command: Pig
2. X = Load the data set using Pig Storage;
3. Y = For each X generate the crime by year;
4. Z = Group by year;
5. Data = For each Z generate group, SUM(X.year);
6. Store output.

5.3 Total number of Crimes in 2012 by Types

Input: Crime data set

Output: Total number of crimes with IUCR code

Algorithm used in Grunt Shell:

1. Enter into the Grunt shell using command: Pig
2. X = Load the data set using Pig Storage;
3. Y = For each X generate the crime by year;
4. Z = Group by year;
5. Data = For each Z generate group, SUM(X.year);
6. Store output.

5.4 Total number of Crimes in 2012 by Description

Input: Crime data set

Output: Total number of crimes with IUCR code

Algorithm used in Grunt Shell:

1. Enter into the Grunt shell using command: Pig
2. X = Load the data set using Pig Storage;
3. Y = For each X generate the crime by year;
4. Z = Group by year;
5. Data = For each Z generate group, SUM(X.year);
6. Store output.

6. SCOPE

Limitations of this analysis are:

- Only four nodes are used in distributed cluster to process the data. We can use many more nodes for fast processing.
- Only four types of analysis are done in this case

study but we can discover many such facts or pattern from this dataset.

- There could be many more techniques and area of application to discover new trends but as an author we cannot claim for thorough research.

7. CONCLUSION AND FUTURE WORK

Big data analytics is used to transform huge amount of raw data into meaningful data which will helps in decision support system. With increase in population and crime rates huge amount of raw data is generating, only need is to process that data in order to find some crime trends. Extra security must be provided to the society so that law and order can be maintained properly and there is a sense of safety among citizens of country. Hadoop is perfectly suitable framework for such applications where huge amount of data is present. Pig Script gives an extra advantage over Map-reduce programming with its simplicity of writing script.

7.1 Future Work

- Similar kind of analysis can be done in various sectors and in different applications like social media analysis, Baking sector, Transportation sector etc.

- This analysis can be further carried out on number of clusters

8. ACKNOWLEDGEMENT

We authors would like to thank pimpri chinchwad college of engineering, pune for providing required aid to carry out the research successfully.

9. REFERENCES

[1] K. Goodhope, J. Koshy, et al, Building LinkedIn's Real-time Activity Data Pipeline, Data Engineering, volume 35, issue 2, pp. 33-45, 2012.

[2] Yeonhee Lee and Youngseok Lee, Toward scalable internet traffic measurement and analysis with Hadoop, ACM SIGCOMM Computer Communication, 2013, volume 43, issue 1.

[3] Doug Howe, Maria Costanzo, Petra Fey, Takashi Gojobori, Linda Hannick, Winston Hide, David P. Hill, Renate Kania, Mary Schaeffer, Susan St Pierre, Simon Twigger, Owen White and Seung Yon Rhee , Big data: The future of biocuration, Nature, international weekly journal of science 455, 47-50,4 September 2008

[4] Clifford Lynch, Big data: How do your data grow?, Nature , international weekly journal of science ,455, 28-29

[5] Adam Jacobs, The pathologies of big data, Communications of the ACM - A Blind Person's Interaction with Technology ,Volume 52 Issue 8, August 2009

[6] Min Chen, Shiwen Mao and Yunhao Liu, Big Data: A Survey, Springer- Mobile Networks and Applications, Volume 19, Issue 2, pp 171-209, 2014

[7] Alexandros Labrinidis and H. V. Jagadish, Challenges and opportunities with big data, ACM- Proceedings of the VLDB Endowment, Volume 5 Issue 12, August 2012

[8] Shadi Ibrahim, Hai Jin, Lu Lu, Li Qi, Song Wu, and Xuanhua Shi, Evaluating MapReduce on Virtual Machines: The Hadoop Case, Springer: Cloud Computing Lecture Notes in Computer Science, Volume 5931, 2009, pp 519-528

[9] Lecture Notes in Computer Science, 2013.

[10] www.edureka.co/

[11] https://www.wikipedia.org/

[12] http://www.guru99.com/introduction-to-pig-and hive.html

[13] .http://tutorialshadoop.com/pig-interview-questions-answers-part-3/

[14] Aggarwal Sonal, and Vishal Bhatnagar, Technological applications of data mining and virtual reality: a literature survey and classification, International Journal of Intercultural Information Management, 2013.

[15] Ngai, E.W.T, Application of data mining techniques in customer relationship management: A literature review and

[16] classification", Expert Systems With Applications, 200903

Bhardwaj, Vibha, and Rahul Johari, Big data analysis: Issues and challenges, 2015 International Conference on Electrical Electronics Signals Communication and Optimization (EESCO), 2015.

[17] Ding, Zhiyang, Xunfei Jiang, Shu Yin, Xiao Qin, Kai-Hsiung Chang, Xiaojun Ruan, Mohammed I. Alghamdi, and Meikang Qiu, Multicore-Enabled Smart Storage for Clusters, 2012 IEEE International Conference on Cluster Computing, 2012.