Analysis of Boston's Crime Data using Apache Pig

V.C Chandra Kishore[1], Aditya Shekhar Camarushy[2]

[1]New Horizon College of Engineering, Bengaluru, Karnataka, India
[2]Manipal Institute of Technology, Manipal, Karnataka, India

---

***Abstract*—** Big Data is a technology development happened to the hyperbolic rate of data growth, advanced new data varieties and parallel advancements in technology stake. The types of data could be structured, unstructured or semi-structured, making if difficult to analyse using the standard data management strategies. Hadoop is a framework for the analysis and transformation of incredibly giant data sets using MapReduce. Pig is an apache open source project which is used to reduce the difficulties of coding Map Reduce applications using Pig Latin which is an SQL like query language. In this paper we introduce the Big data analytics tools like apache pig and also work on a analysing a crime data set of Boston.

Keywords-Big Data, Hadoop, Apache Pig, analysis, crime data

I. INTRODUCTION

The term Big Data refers to high volume of data that can't be processed or managed by traditional database systems due to its size and complexities. In today's world data is generated from various sources, the size of this data is increasing at a rapid phase and is exceeding the size of an Exabyte. [1] Moreover with data being generated the computer systems are much faster yet analysing of such massive data sets is a challenge. [2] The data could be generated from various sources across internet; social media is just one of them. The size of Big data is massive and it keeps increasing, 90% of all data today was generated in the last two years.

To Illustrate this consider the data generated around us every minute of the day [3]

SOURCE	AMOUNT OF DATA
Uber Riders	take 45,787.54 trips
Google	conducts 3,607,080 searches
Netflix users	stream 69,444 HRS of video
Amazon	makes $258,751.90 in sales

Fig. 1 How much data does the world generate every minute

This is a huge amount of data that is being generate every minute. This data that is generated is used for making better decisions at a company, in other words Netflix has seen 10% decrees in the users watch time after the new update of their app which tells them to review their new features: all this is made in order to get the most profits possible for any business.

II. TYPES OF BIG DATA

Big data can be explained in accordance to the 3 V's mentioned by Gartner (2012) [4] as velocity, variety and volume and this is explained in detail below:

1. *Velocity*

The Velocity is the speed at which the information is created, stored and analysed. The speed at which the information is generated is unbelievable for each minute we tend to transfer a hundred hours of video on YouTube, over two hundred million emails are sent, around twenty million photos are viewed and nearly three hundred thousand tweets are sent. The huge data generated from various sources continues to grow.

2. *Volume*

It refers to unimaginable amounts of data that's being generated every second from social media, mobile phones, cars, sensors, videos. etc. The amount of information within the world doubles every 2 years. Self-driving cars alone generates two Petabyte of data per annum. [5] Whether the information is considered as huge information or not, relies upon volume of the information. It would be tough for us to manage such Brobdingnagian amounts of data in the past however with the current decrease in storage prices, superior storage solutions like Hadoop and Scala it is not an issue.

3. *Variety*

It refers to the varied formats within which data is formed. In the past all the information that we possessed was structured data which was neatly fitted in columns and rows however today, ninety percent of the information that's generated an the organisation is unstructured data. In general, the various forms of data/information are:

1. Structured data: Relational data

2. Semi-structured data: XML, E-mail data

3. Unstructured data: Word, PDF, Text, Media Logs.

III. HADOOP

Hadoop is an open source, distributed framework developed and maintained by the Apache Foundation which is written in java. Hadoop's [6] MapReduce processes the information and HDFS stores the giant datasets in a cluster. It is employed to handle giant and sophisticated data which can be structured, unstructured or semi-structured that doesn't match into tables.

HDFS (Hadoop Distributed File System)-used for storage. HDFS supports a conventional hierarchal file organization. A user or an application will produce directories and store files within these directories. The filing system namespace hierarchy is analogous to most different existing file systems [7]. HDFS has the many nodes namely: Name node, Secondary name node, Data node and the Map Reduce which is used for processing has Job Tracker and Task Tracker.

a) PIG

Apache Pig is a platform for analysing giant data sets that consists of a high-level language for expressing data analysis programs, including infrastructure for evaluating these programs. [8] It is designed to reduce the complexities to code a map-reduce application. 10 lines of code in Pig is equivalent to 100 lines of code in Java, It is used to load the data, apply various filters on it and finally dump the data in a format as per users requirement.

Moreover, to analyse data using Pig, users need to write scripts using Pig Latin language. It is a Procedural Data Flow Language used by Researchers and Programmers but does not have a dedicated metadata database.

b) Hive

After gathering the data into HDFS they are then analysed using Hive queries. Hive [9] data warehouse software system which facilitates querying and managing massive datasets located in distributed storage. Initially it was developed by Facebook later the Apache Foundation took it and developed it further as open source. It is used by many companies, one such example is Amazon where hive is used in their Elastic MapReduce.

To make it clear Hive is not a relational database, it is designed for Online Transaction Processing. It stores schema in the database and processed data into HDFS also it provides SQL type language called HiveQL.

IV. PIG ARCHITECTURE AND COMPONENTS

Pig architecture consists of Pig Latin Interpreter and it'll be executed on consumer/client Machine. It uses Pig Latin scripts and it converts the script into a series of map-reduce jobs[10]. Later it executes the map-reduce jobs and saves the output result into HDFS. In between, it performs completely different operations like Parse, Compile, Optimize and plan the Execution of the data.

Apache Pig has a component referred to Pig Engine which takes the Pig Latin Scripts as an input and converts those scripts into Map-reduce Jobs. The various Apache Pig Components are:

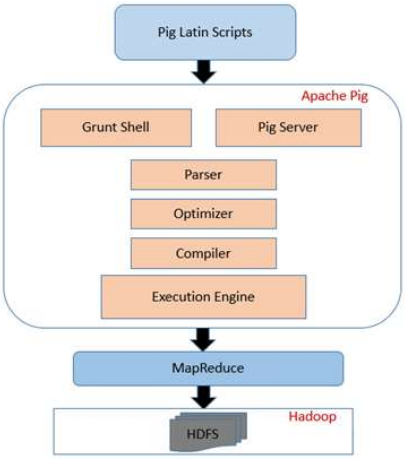

Fig. 2 Architecture of Pig

- Parser

It is used for checking the syntax of the script, type verifying and other various checks. The output of this component will be a directed acyclic graph (DAG), which represents the PL Statements and operators. The DAG and the operators if the script is represented as nodes, data flow is represented as edges.

- Optimizer

The DAG generated from the parser is then passed to the optimizer which performs various logical optimizations such as projections.

- Compiler

In this the compiler compiles the previously optimized logical plan into a series of Map-reduce Jobs for execution.

- Execution engine

Finally, these Jobs are then submitted Hadoop in a order and are executed on Hadoop to produce the desired results.

Pig Execution Modes

Pig will run in 2 execution modes. The modes depend on where the Pig script goes to run and where the info is residing. The data may be kept on a single machine, i.e. Local File System or it may be kept in a distributed system like typical Hadoop Cluster environment.

The execution modes of Pig are:

- *Local Mode*

Here the files are present and processed from local host and local file system. There is no need for Hadoop or HDFS. This mode is usually used for testing purpose. Once executed it provides output on top of LFS. Pig runs in a single JVM and accesses the native/local file system.

To start local mode of execution:

```
pig -x local
```

- *MapReduce mode*

In this mode of execution, we work with data that is present in the HDFS using Pig. Here we use Pig Latin scripts to process the data and a Map-reduce job is invoked in the back-end.

To start MapReduce mode of execution:

```
pig -x MapReduce
or
pig
```

V. PIG LATIN

It is a programming language which is used to create programs that run on Hadoop

Pig Latin data model is fully nested kind and supports advanced non-atomic data varieties like maps and tuple. Below is a Pig Latin data model.

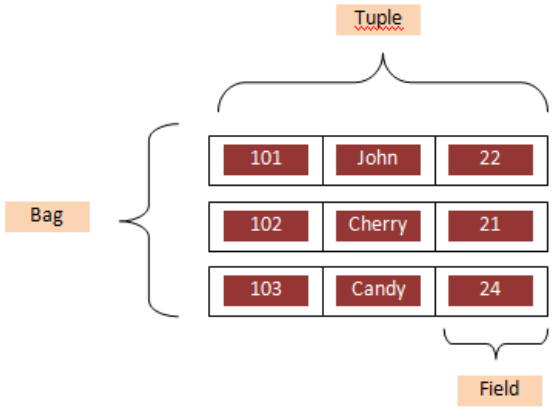

Fig. 3 Pig Latin Data model

1. *Atom*
   - It is a single value any data type in Pig Latin.
   - It is in the form of string and can be used as a string or a number.
   - example: 'Raju' or 39
2. *Tuple*
   - It is a record consisting of an ordered set of fields and these fields can be of any type.
   - It is similar to that of a row in RDBMS
   - example: (Chandra, 20)
3. *Bag*
   - It is a collection of unordered tuples
   - In a bag each tuple could have any number of fields
   - It is represented by { }
   - example: {(Anju, 20),(Manu, 22)}
4. *Map*
   - It is a set of key-value pairs where the key has to be unique and the value could be of any type.
   - It is denoted by [ ]
   - example: [name#John, age#22]

Data types

S. No	Data Type	Description
1	int	Represents a signed 32-bit integer. Example: 5
2	float	Represents a signed 32-bit floating point. Example: 6.5F
3	double	Represents a 64-bit floating point. Example: 14.5
4	chararray	Represents a character array (string) in Unicode UTF-8 format. Example: 'Chandra'
5	Boolean	Represents a Boolean value. Example: true/ false.
6	Datetime	Represents a date-time. Example: 970-01-01T00:00:00.000+00:00
7	Big decimal	Represents a Java Big Decimal Example 185.98376256272893883

Fig. 4 Data Types of Pig Latin

VI. PIG COMMANDS

1. Loading the data from LFS/HDFS to Pig

To put a file into Hadoop

```
hadoop fs -put file /user/cloudera
```

To switch to local execution mode in PIG

```
pig -x local
```

To load the data file into PIG

```
bag2 = load '/home/cloudera/file' using PigStorage(',') as (id:int,name:chararray);
```

To see the content of the bag

```
dump bag2;
```

2. Utility commands

Clear is used to clear the screen of the Grunt shell.

```
grunt> clear
```

Help is used to view the list of pig commands and properties

```
grunt> help
```

Dump is used to view the results on the screen

```
grunt> dump rel_name
```

Quit command is to quit the Grunt shell

```
grunt> quit
```

Describe is used to view the schema

```
grunt> describe rel_name
```

3. Filtering

There are three operators used for filtering they are:

- FILTER Operator

- FOREACH Operator

- Distinct Operator

Filter is used to select specific tuples from a relation based on requirement. It is similar to where statement in MySQL.

Syntax

grunt> Rel_name = FILTER Rel2_name BY (condition);

example - filters the emp whose experience is more than 10 years

```
a = filter file by exp>10;
dump a;
```

For each is used to generate data transformation based on columns. It is similar to select statement in MySQL.

Syntax

grunt> Rel_name = FOREACH Rel2_name GENERATE (required data);

example - generate each emp name and emp office

```
grunt> a = foreach file generate name,office;
grunt> dump a;
```

VII. Proposed Methodology

Steps to be followed are:

1. Collect the crime data of Boston from various trusted web sources.

2. After collection of the data sets load the crime data of Boston using Hadoop command line.

3. Store the data in Hadoop Distributed File System as It is appropriate for those applications which have massive data sets.

4. The crime data of Boston is then processed by the Map-reduce which is the processing engine of Hadoop.

5. Analyse the data with the help of other Big Data tools which works on top of Hadoop and process the data

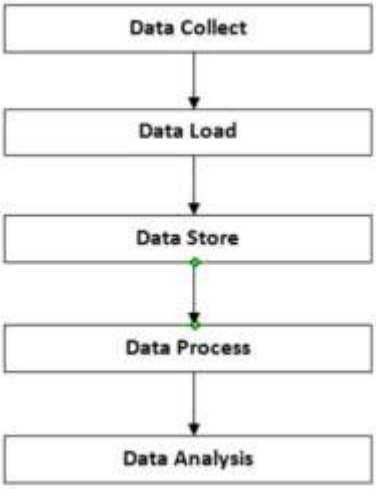

Fig. 5 Methodology for analysing the data

VIII. Research Questions

1. Find the top 3 reporting areas in Boston.
2. Find the top 5 STREETS safest streets in-terms of number of crimes registered.
3. Find the top 3 most active crime hours of the day
4. Find the number of crimes happened on Saturday and Sunday of the week.
5. Count the top 3 offence code groups in Boston.
6. Find the year which had the highest number of crimes in the past 3 years

IX. Experimental Findings

1. Find the top 3 STREETS safest streets in-terms of number of crimes registered.

```
a = group crime by REPORTING_AREA;

b = foreach a generate group, COUNT_STAR (crime.REPORTING_AREA) as no;

c = order b by no desc;

d = limit c 3;
```

```
(,26962)
(111,2264)
(186,1982)
```

[empty]	6%
111	1%
186	1%
329	1%
Other (875)	92%

Valid	319k	100%
Mismatched	0	0%
Missing	0	0%
Unique	879	
Most Common		6%

Fig. 6 Output and explanation of Query 1

2. Find the top 5 STREETS safest streets in-terms of number of crimes registered.

```
a = group crime by STREET;

b = foreach a generate group, COUNT_STAR (crime.STREET) as num;

c = order b by num asc;

d = limit c 5;
```

```
(FRANCESCA ST,1)
(WEDGEWOOD RD,1)
(MYSTIC,1)
(FISH PIER RD,1)
(N CRES,1)
```

Valid	308k	97%
Mismatched	0	0%
Missing	10.9k	3%
Unique	4657	
Most Common	WASHINGTON	4%

Fig. 7 Output and explanation of Query 2

3. Find the top 3 most active crime hours of the day

```
e= foreach crime generate SUBSTRING (OCCURRED_ON_DATE,9,11);

f = group e by times;

g = foreach f generate group as time,COUNT_STAR (e.times) as no;

res = limit (order g by no desc ) 3;
```

```
(17,19904)
(18,19432)
(16,19289)
```

Valid	319k	100%
Mismatched	0	0%
Missing	0	0%
Mean	13.1	
Std. Deviation	6.29	
Quantiles	0	Min
	9	25%
	14	50%
	18	75%
	23	Max

Fig. 8 Output and explanation of Query 3

4. Find the number of crimes happened on Saturday and Sunday of the week.

```
a = group crime by DAY_OF_WEEK;

b = foreach a generate group,COUNT_STAR (crime.INCIDENT_NUMBER) as no;

c = filter b by group=='Saturday' or group=='Sunday';
```

```
(Sunday,39603)
(Saturday,43786)
```

Friday	15%
Wednesday	15%
Thursday	15%
Tuesday	15%
Other (3)	41%

Valid	319k	100%
Mismatched	0	0%
Missing	0	0%
Unique	7	
Most Common	Friday	15%

Fig. 9 Output and explanation of Query 4

5. Count the top 3 offence code groups in Boston.

```
a = group crime by OFFENSE_CODE_GROUP;

b = foreach a generate group,COUNT_STAR (crime.OFFENSE_CODE_GROUP) as no;

c = limit (order b by no desc) 5;
```

```
(Motor Vehicle Accident Response,37132)
(Larceny,25935)
(Medical Assistance,23540)
(Investigate Person,18750)
(Other,18075)
```

Motor Vehicle Accident Response	12%
Larceny	8%
Medical Assistance	7%
Investigate Person	6%
Other (63)	67%

Fig. 10 Output and explanation of Query 5

6. Find the year which had the highest number of crimes in the past 3 years

```
a = group crime by YEAR;

b = foreach a generate group,COUNT_STAR (crime.INCIDENT_NUMBER) as no;

c = filter b by group=='2018' or group=='2017' or group =='2016';

d = limit (order c by no desc) 1;
```

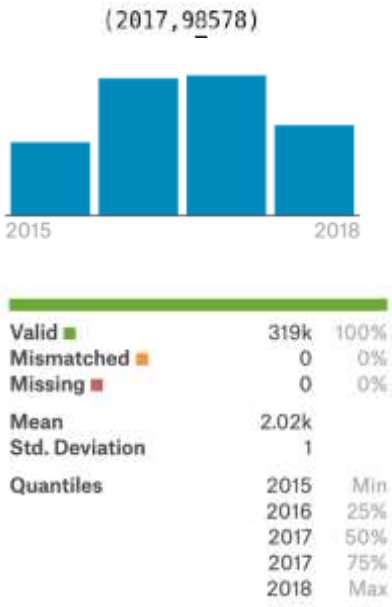

Fig. 11 Output and explanation of Query 6

X. Conclusion

Hadoop MapReduce is currently preferred for large-scale data analytics. Big-data analytics with pig and hive shows the important problems faced by customers and helps the companies to rectify these issues. This paper discusses the concepts of Hadoop and provides explanation for basic commands to analysing the crime data of Boston. The insights gathered from this study could be used to improve the current situation of Boston with regards to the crimes.

References

[1] Peter Charles, Nathan Good, Laheem Lamar Jordan, how- much-info-2003

[2] Xu R, Wunsch D. Clustering. Hoboken: Wiley-IEEE Press; 2009.

[3] How much data does the world generate every minute" Available [https://www.iflscience.com/technology/how-much-data-does-the-world-generate-every-minute/]

[4] Mark van Rijmenam " Self driving cars create 2 petabytes data" from datafloq.com

[5] http://hadoop.apache.org/

[6] https://hadoop.apache.org/docs/r1.2.1/hdfs_design.html

[7] https://pig.apache.org/

[8] https://hive.apache.org/

[9] http://a4academics.com/tutorials/83-hadoop/837-hadoop-pig