A case study on crime dataset of "city of Baltimore, Maryland(USA)" using Big Data Analytics with pig

Akash Chowrasia

B.Tech Scholar, Department of Computer Science & Engineering,

Amritsar College of Engineering & Technology,

Amritsar, Punjab

chowrasia.akash08@gmail.com

Rohit Prasad

B.Tech Scholar, Department of Computer Science & Engineering,

Amritsar College of Engineering & Technology,

Amritsar, Punjab

rohitprasad25061999@gmail.com

Raghav Vashisht

Department of Computer Science & Engineering,

Amritsar College of Engineering & Technology,

Amritsar, Punjab

raghavvashisht194@gmail.com

Abstract:- In the information era, enormous amounts of data have become available on hand of data analysts to analyze in the form of datasets which are not only big, but also high in variety and velocity, using these analysis reports Now, human society is making many types of counter measures in the field of medical, weather, crime and many more. The case study on "crime dataset" found on keggle.com specifying some threats and wanted places in "city of Baltimore, Maryland(USA)" where, there is a need of some extra counter measures against some criminal activity.

Keywords:- Big data analysis, Apache pig, Hadoop Distributed file system.

1. TOOLS AND TECHNOLOGYS USED

Imagine a world without data storage; a place where every detail about a person or organization, every transaction performed, or every aspect which can be documented is lost directly after use. It will lead to a disaster on human society like crime, diseases, governments and all the organization's.

Due to data storage and it's analysis the structure of whole human society is now changed. We are capturing each and every data generated by human activity. By analyzing these data's all the needs of human is being completed every day like city structure, safety measures, traffic control, medical improvements, human comfort, education quality, crime control etc.

Now a days we are generating enormous amounts of data every minute which are generating difficulties to store and analyze them. To store and retrieve very efficiently Big Data Analytics and its tools are gathering huge trust between many big organizations.

1.1 Big Data Analytics

Big Data is a term used to describe a collection of data that is huge in size and yet growing exponentially with time. In short such data is so large and complex that none of the traditional data management tools are able to store it or process it efficiently.

1.1.1 Types of Big Data

Big Data could be found in three forms:

1. Structured 2. Unstructured 3. Semi-structured

Structured :- Any data that can be stored, accessed and processed in the form of fixed format is termed as a 'structured' data. Over the period of time, talent in computer science has achieved greater success in developing techniques for working with such kind of data (where the format is well known in advance) and also deriving value out of it. However, nowadays, we are foreseeing issues when a size of such data grows to a huge extent, typical sizes are being in the rage of multiple zettabytes.

For example:- Record of employees.

Unstructured:- Any data with unknown form or the structure is classified as unstructured data. In addition to the size being huge, un-structured data poses multiple challenges in terms of its processing for deriving value out of it. A typical example of unstructured data is a heterogeneous data source containing a combination of simple text files, images, videos etc. Now day organizations have wealth of data available with them but unfortunately, they don't know how to derive value out of it since this data is in its raw form or unstructured format.

For example:- Results of Google search.

Semi-Structured:- Semi-structured data can contain both the forms of data. We can see semi-structured data as a structured in form but it is actually not defined with e.g. a table definition in relational DBMS.

For example:- Personal data stored in an XML file

1.1.2 Characteristics of Big Data

(i) **Volume –** The name Big Data itself is related to a size which is enormous. Size of data plays a very crucial role in determining value out of data. Also, whether a particular data can actually be considered as a Big Data or not, is dependent upon the volume of data. Hence, 'Volume' is one characteristic which needs to be considered while dealing with Big Data.

(ii) **Variety –** The next aspect of Big Data is its variety. Variety refers to heterogeneous sources and the nature of data, both structured and unstructured. During earlier days, spread sheets and databases were the only sources of data considered by most of the applications. Nowadays, data in the form of emails, photos, videos, monitoring devices, PDFs, audio, etc. are also being considered in the analysis applications. This variety of unstructured data poses certain issues for storage, mining and analyzing data.

(iii) **Velocity –** The term 'velocity' refers to the speed of generation of data. How fast the data is generated and processed to meet the demands, determines real potential in the data. Big Data Velocity deals with the speed at which data flows in from sources like business processes, application logs, networks, and social media sites, sensors, Mobile devices, etc. The flow of data is massive and continuous.

(iv) **Variability –** This refers to the inconsistency which can be shown by the data at times, thus hampering the process of being able to handle and manage the data effectively.

(v) **Value -** How will the extraction of data work? Here, our fourth V comes in, which deals with a mechanism to bring out the correct meaning out of data. First of all, you need to mine the data, i.e., a process to turn raw data into useful data. Then, an analysis is done on the data that you have cleaned or retrieved out of the raw data. Then, you need

to make sure whatever analysis you have done benefits your business such as in finding out insights, results, etc. which were not possible earlier.

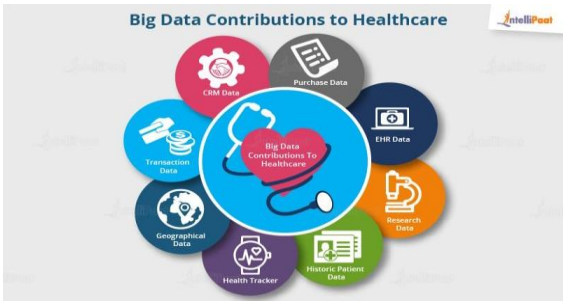
1.1.3 Applications of Big Data Analytics

Big Data technologies can be used for creating a staging area or landing zone for new data before identifying what data should be moved to the data warehouse. In addition, such integration of Big Data technologies and data warehouse helps an organization to offload infrequently accessed data.

Big Data in Education

Education industry is flooding with huge amounts of data related to students, faculty, courses, results, and what not. Now, we have realized that proper study and analysis of this data can provide insights which can be used to improve the operational effectiveness and working of educational institutes.

Big Data in Healthcare



Healthcare is yet another industry which is bound to generate a huge amount of data. Following are some of the ways in which big data has contributed to healthcare:

- Big data reduces costs of treatment since there is less chances of having to perform unnecessary diagnosis.
- It helps in predicting outbreaks of epidemics and also in deciding what preventive measures could be taken to minimize the effects of the same.

Big Data in Government Sector

Governments, be it of any country, come face to face with a very huge amount of data on almost daily basis. The reason for this is, they have to keep track of various records and databases regarding their citizens, their growth, energy resources, geographical surveys, and many more. All this data contributes to big data.

Big Data in Media and Entertainment Industry

With people having access to various digital gadgets, generation of large amount of data is inevitable and this is the main cause of the rise in big data in media and entertainment industry. Other than this, social media platforms are another way in which huge amount of data is being generated. Although, businesses in the media and entertainment industry have realized the importance of this data, and they have been able to benefit from it for their growth.

1.2 HADOOP

Hadoop is an Apache open source framework written in java that allows distributed processing of large datasets across clusters of computers using simple programming models. The Hadoop framework application works in an environment that provides

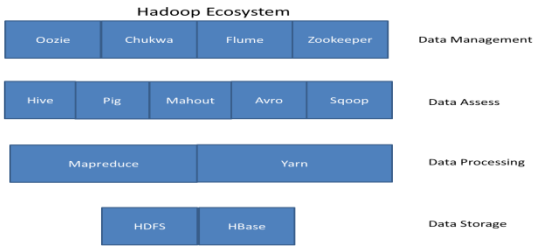
distributed storage and computation across clusters of computers. Hadoop is designed to scale up from single server to thousands of machines, each offering local computation and storage.

1.2.1 Hadoop Ecosystem

Hadoop Ecosystem is a platform or a suite which provides various services to solve the big data problems. It includes Apache projects and various commercial tools and solutions. There are four major elements of Hadoop i.e. HDFS, MapReduce, YARN, and Hadoop Common. Most of the tools or solutions are used to supplement or support these major elements. All these tools work collectively to provide services such as absorption, analysis, storage and maintenance of data etc.

Following are the components that collectively form a Hadoop ecosystem:

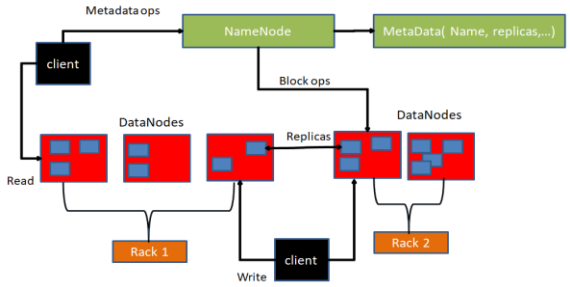
- **HDFS:** Hadoop Distributed File System
- **YARN:** Yet Another Resource Negotiator
- **MapReduce:** Programming based Data Processing
- **Spark:** In-Memory data processing
- **PIG, HIVE:** Query based processing of data services
- **HBase:** NoSQL Database
- **Mahout, Spark MLLib:** Machine Learning algorithm libraries
- **Solar, Lucene:** Searching and Indexing
- **Zookeeper:** Managing cluster
- **Oozie:** Job Scheduling



1.2.2 HDFS Architecture

Hadoop HDFS architecture is a master/slave like Architecture in which master is NameNode and slave is DataNode. HDFS Architecture consist of single NameNode and Multiple DataNodes.

- In today's scenario we are dealing with a huge amount of dynamic data which is not possible to handle with local File system.
- These huge amount of Data which we have to analyze every second is known as Big Data. The volume of Big Data is increasing with a high velocity over per second.
- Big Data files are always at least 1 TB in size.
- Hadoop distributed file system (HDFS) is the worlds most Trusted Big Data storage system which stores very large files running on a cluster of "cheap" commodity hardware.



The components of the HDFS Architecture are as follows..

NAMENODE :

NameNode in HDFS Architecture is the master of the cluster. NameNode stores meta-data rather than the actual data. Meta-data is nothing but the information of the actual data like data block information, replica's information and Rack information etc. NameNode manages all the operations on each blocks of data as it is the only one who knows all the information about the each data blocks of each file. This is the Single Point Of Failure (SPOF) for the entire HDFS Architecture Because it is the only source for retrieval of Data, if it goes down whole the HDFS cluster goes down.

Feature's of NameNode:

- NameNode is responsible for the management of file system namespace/meta-data/file blocks. Here are some important **file's** related to namespace of NameNode:
  - **FsImage** – It is an "Image file". FsImage contains entire file system and stores as a file in the NameNode's local file system.
  - **EditLogs** – NameNode directly does not modifies any changes made in FsImage file. EditLogs contains all the recent modifications made to the file system on the most recent FsImage.

NameNode receives a create/update/delete request from client After that this request is recorded to edits file.

- It runs on 1 machine to several machines.
- It is single point of Failure.
- It manipulates all the file system operations like naming, opening, closing of files/directories.
- It Handles and verify the access of files to the client's.
- It is responsible for the High Availability of Data on the basis of **Replica Placement Policy**.
- It is responsible to ensure that all the DataNodes are alive or not, By the help of **Heartbeats** and **Block report**.
- It is responsible for the management of replication factor of all the blocks.

SECONDARY NAMENODE :

Secondary NameNode helps NameNode to perform house-keeping tasks of the cluster. It is not used for high Availability or a backup for the NameNode, it is just for speeding-up the process of NameNode. It requires similar hardware as NameNode.

Feature's of Secondary NameNode:

- In HDFS when daemons starts, Secondary NameNode loads the FsImage and EditLogs from the NameNode. And then merges EditLogs with the FsImage. It keeps edits log size within a limit. It stores the modified FsImage into persistent storage. And we can use it in the case of NameNode failure. These operations takes ample amount of time. If This operation is done by NameNode then it will take a lot of time of Namenode.
- All the operation performed by NameNode is always stored in physical memory. Secondary Namenode is set at a regular checkpoint, where it stores all the updates to the permanent memory at a regular interval of time.

DATANODES :

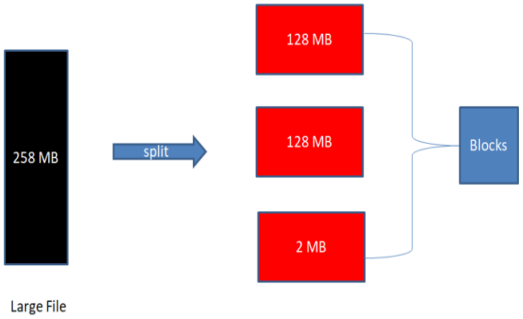
DataNode is the Slave of the HDFS architecture. It stores the actual data in HDFS. it is responsible for the read and write operations as per the request of the clients. they can deploy on cheap commodity hardwires like ext3 and ext4.

Feature's of DataNode:

- They are responsible for storing and retrieving of Data blocks.
- They Runs on many system.
- Block creation, deletion and replication are done by DataNodes as per instruction of NameNode.
- They are arranged in the form of Racks.
- DataNode sends heartbeat message to the NameNode to report the health of HDFS. By default, this frequency is set to 3 seconds.

BLOCKS :

In HDFS, NameNode splits huge files into small files known as Blocks. all the operations and manipulations on the blocks are done by NameNode. All the blocks of a particular file are stored into different DataNodes. The default **size** of each block is 128 MB, which can be configured as per individual's need. All the blocks are having same size except the last block. The last block of a file can be of a size less than or equal to 128 MB. For example, if we are storing a file of size 258 MB, then this file is splits into 3 blocks 2 of them are of size 128 MB and the last block is of the size 2 MB.

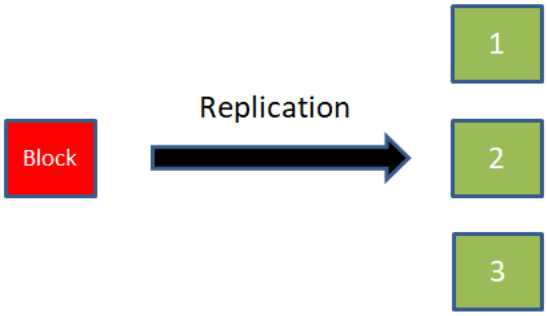


Feature's of Blocks:

- Blocks are managed by NameNode and stored by DataNode.
- Every blocks are independent of each other. they do not have any information about each other.
- These are the smallest unit of file which can be stored in any file system
- The default size of the blocks are 128 MB which can be configured as per requirement
- The size of each blocks are equal except the last block
- The target for block management is to minimize the cost of seeks as compare to transfer rate.
- It provides the seek time of 1% transfer rate when the block size is at least of 100 MB.

REPLICATION :

NameNode stores the same copy of a block into multiple DataNodes. The number of copies of same block is termed as number of replicas and the process of storing multiple copies of same block is known as replication of a block. If any DataNode gets failed then DataNode pick-up the block from the other DataNode. The number of replicas of a file in HDFS is called replication factor. For example if replication factor is 3 and the block size is 128 then each block will end up by occupying a space of 384 MB (3*128 MB) for each block.

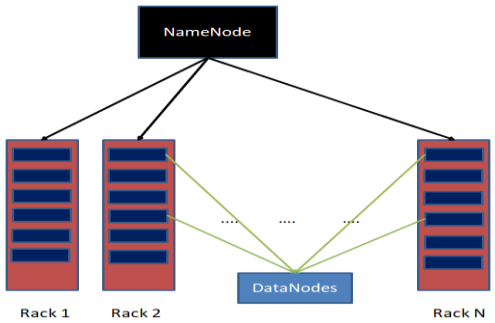


Feature's of Replication :

- Replication provides the feature of **fault tolerance**.
- The default replication factor is 3.
- DataNode sends block report to the NameNode periodically to maintain the replication factor.
- When a block is over-replicated (number of replications of a block are more than the replication factor), then NameNode deletes the extra replica of the block.
- When the block is under-replicated (number of replication of a block are less than the replication factor), then NameNode adds the required number of replicas.

RACK AWARENESS :

In a large Hadoop cluster the term Rack awareness is introduced to improve the high availability of Data and performance of the entire cluster. Rack awareness term provides an algorithm to the NameNode known as Rack Awareness Algorithm, using which NameNode maintains a huge amount of Data blocks and its replicas in Rack like structure in which every Rack consists of a specific number of DataNode.



Feature's of Rack Awareness :

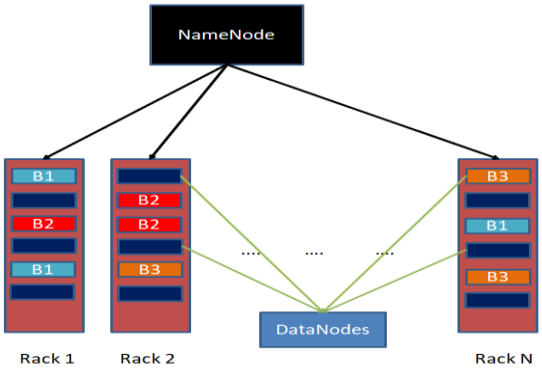
- For reading/writing operation NameNode chooses the DataNode from the same Rack or the nearby rack to improve the traffic and speed of the cluster.
- NameNode maintains Rack id's to get access to each rack and to perform various operations over there.
- In large Hadoop cluster Rack awareness is responsible for less complexity of read/write pipeline, due to which there is less chances of Data getting lost.

REPLICA PLACEMENT POLICY :

NameNode determines the replica placement of blocks. According to Replica Placement policy if Replication Factor is 3 then…

- First Replica is stored on the local Rack.
- Second replica is stored on the local Rack but in different DataNode.
- Third replica is stored on the different Rack.

For example, we have a huge file B. To store this file first NameNode split's it into small blocks, suppose it splited into 3 blocks B1, B2 and B3. Now according to Replica Placement Policy it will be placed into cluster as follows :



Feature's of Replica Placement Policy:

- It usage different DataNodes and Different Racks also.
- If DataNode gets failed then NameNode reads the block from different DataNode but if Rack Fails then NameNode reads Block from different Rack.
- It will improve the High Availability of Data to the client.

1.2.3 HDFS commands

HDFS commands using which you can access the Hadoop File System :

- **fsck**

HDFS Command to check the health of the Hadoop file system.

Command*:* hdfs fsck /

- **ls**

HDFS Command to display the list of Files and Directories in HDFS.

Command*:* hdfs dfs –ls /

- **mkdir**

HDFS Command to create the directory in HDFS.

Usage: hdfs dfs –mkdir /directory_name

Command*:* hdfs dfs –mkdir /mydir

- **cat**

HDFS Command that reads a file on HDFS and prints the content of that file to the standard output.

Usage*:* hdfs dfs –cat /path/to/file_in_hdfs

Command: hdfs dfs –cat /mydir/mydata

- **copyFromLocal**

HDFS Command to copy the file from a Local file system to HDFS.

Usage: hdfs dfs -copyFromLocal<localsrc><hdfs destination>

Command: hdfs dfs –copyFromLocal /home/test /mydir

- **copyToLocal**

HDFS Command to copy the file from HDFS to Local File System.

Usage: hdfs dfs -copyToLocal<hdfs source><localdst>

Command: hdfs dfs –copyToLocal /mydir/test /home/

- **put**

HDFS Command to copy single source or multiple sources from local file system to the destination file system.

Usage: hdfs dfs -put <localsrc><destination>

Command: hdfs dfs –put /home/test /user

- **get**

HDFS Command to copy files from hdfs to the local file system.

Usage: hdfs dfs -get <src><localdst>

Command: hdfs dfs –get /user/test /home

- **count**

HDFS Command to count the number of directories, files, and bytes under the paths that match the specified file pattern.

Usage: hdfs dfs -count <path>

Command: hdfs dfs –count /user

- **rm**

HDFS Command to remove the file from HDFS.

Usage: hdfs dfs –rm <path>

Command: hdfs dfs –rm /newdir/test

- **cp**

HDFS Command to copy files from source to destination. This command allows multiple sources as well, in which case the destination must be a directory.

Usage: hdfs dfs -cp<src><dest>

Command: hdfs dfs -cp /user/hadoop/file1 /user/hadoop/file2

- **mv**

HDFS Command to move files from source to destination. This command allows multiple sources as well, in which case the destination needs to be a directory.

Usage: hdfs dfs -mv <src><dest>

Command: hdfs dfs -mv /user/hadoop/file1 /user/hadoop/file2

- **expunge**

HDFS Command that makes the trash empty.

Command: hdfs dfs -expunge

- **rmdir**

HDFS Command to remove the directory.

Usage: hdfs dfs -rmdir <path>

Command: hdfs dfs –rmdir /user/hadoop

- **help**

HDFS Command that displays help for given command or all commands if none is specified.

Command: hdfs dfs -help

- **usage**

HDFS Command that returns the help for an individual command.

Usage: hdfs dfs -usage <command>

Command: hdfs dfs -usage mkdir

1.3 Apache pig

Pig is a high-level platform or tool which is used to process the large datasets. It provides a high-level of abstraction for processing over the MapReduce. It provides a high-level scripting language, known as Pig Latin which is used to develop the data analysis codes. First, to process the data which is stored in the HDFS, the programmers will write the scripts using the Pig Latin Language. Internally Pig Engine(a component of Apache Pig) converted all these scripts into a specific map and reduce task. But these are not visible to the programmers in order to provide a high-level of abstraction. Pig Latin and Pig Engine are the two main components of the Apache Pig tool. The result of Pig always stored in the HDFS.

Note: Pig Engine has two type of the execution environment i.e. a local execution environment in a single JVM (used when dataset is small in size)and distributed execution environment in a Hadoop Cluster.

Need of Pig: One limitation of MapReduce is that the development cycle is very long. Writing the reducer and mapper, compiling packaging the code, submitting the job and retrieving the output is a time-consuming task. Apache Pig reduces the time of development using the multi-query approach. Also, Pig is beneficial for the programmers who are not from Java background. 200 lines of Java code can be written in only 10 lines using the Pig Latin language. Programmers who have SQL knowledge needed less effort to learn Pig Latin.

Features of Apache Pig:

- For performing several operations Apache Pig provides rich sets of operators like the filters, join, sort, etc.
- Easy to learn, read and write. Especially for SQL-programmer, Apache Pig is a boon.
- Apache Pig is extensible so that you can make your own user-defined functions and process.
- Join operation is easy in Apache Pig.
- Fewer lines of code.
- Apache Pig allows splits in the pipeline.
- The data structure is multivalued, nested and richer.
- Pig can handle the analysis of both structured and unstructured data.

Types of Data Models in Apache Pig:

It consist of the 4 types of data models as follows:

- **Atom**: It is a atomic data value which is used to store as a string. The main use of this model is that it can be used as a number and as well as a string.
- **Tuple**: It is an ordered set of the fields.
- **Bag**: It is a collection of the tuples.
- **Map**: It is a set of key/value pairs.

1.3.1 Apache Pig Architecture

For writing a Pig script, we need Pig Latin language and to execute them, we need an execution environment. The architecture of Apache Pig is shown in the below image.

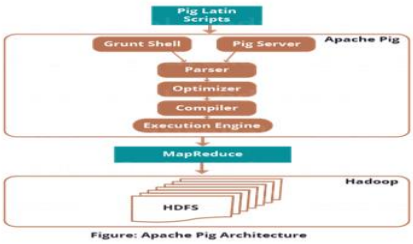


Figure: Apache Pig Architecture

Pig Latin Scripts
Initially as illustrated in the above image, we submit Pig scripts to the Apache Pig execution environment which can be written in Pig Latin using built-in operators.

There are three ways to execute the Pig script:

- ***Grunt Shell*:** This is Pig's interactive shell provided to execute all Pig Scripts.
- ***Script File*:** Write all the Pig commands in a script file and execute the Pig script file. This is executed by the Pig Server.
- ***Embedded Script*:** If some functions are unavailable in built-in operators, we can programmatically create User Defined Functions to bring that functionalities using other languages like Java, Python, Ruby, etc. and embed it in Pig Latin Script file**.** Then, execute that script file.

Parser
From the above image you can see, after passing through Grunt or Pig Server, Pig Scripts are passed to the Parser. The Parser does type checking and checks the syntax of the script. The parser outputs a DAG (directed acyclic graph). DAG represents the Pig Latin statements and logical operators. The logical operators are represented as the nodes and the data flows are represented as edges.

Optimizer

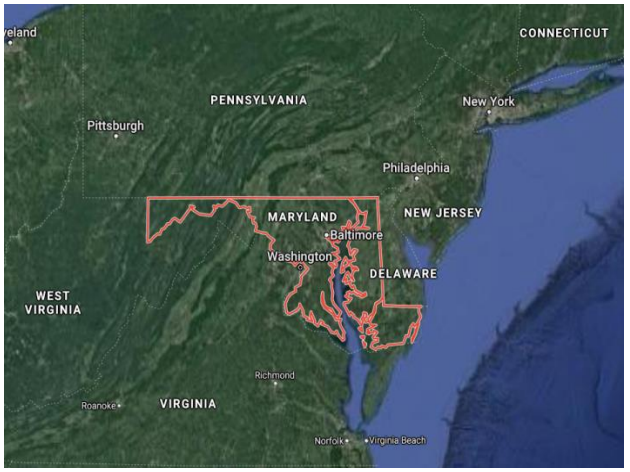
Then the DAG is submitted to the optimizer. The Optimizer performs the optimization activities like split, merge, transform, and reorder operators etc. This optimizer provides the automatic optimization feature to Apache Pig. The optimizer basically aims to reduce the amount of data in the pipeline at any instance of time while processing the extracted data.

Compiler
After the optimization process, the compiler compiles the optimized code into a series of MapReduce jobs. The compiler is the one who is responsible for converting Pig jobs automatically into MapReduce jobs.

Execution engine
Finally, as shown in the figure, these MapReduce jobs are submitted for execution to the execution engine. Then the MapReduce jobs are executed and gives the required result. The result can be displayed on the screen using "**DUMP**" statement and can be stored in the HDFS using "**STORE**" statement.

2. INFORMATION ABOUT DATASET

- Dataset Name: Baltimore.csv
- This is the dataset of the city **"BALTIMORE"** and its nearest places of state 'Maryland' in United-State.



- Total records : 276530
- Total columns: 15
- Column Names:(sno, crime_date, crime_time, crime_code, address, description, Indoor/Outdoor, weapon, post, district, neighbourhood, longitude, attitude, premesis, totalIncidents)
- This dataset contains crime record from 2012-10-01 to 2017-09-02.
- There are total 18 different categories of crime description according to USA rules is involved in the dataset. some of them are as follows:-
    - **Homicide:-** act of one human killing other or the volitional act by any people leads to death of any people. This may result from accident, reckless or negligent acts even if there is no intent to cause harm.

- **Auto-theft:-** robbery of any vehicle.
- **Assault by threat**:- A person commits assault by threat if he or she intentionally threatens another person with imminent bodily injury.
- **Burglary**:- Entering any building against the law intentionally for the act of any crime(even if the crime is not performed) is called burglary.
- **Common assault**:- unlawful violence by any people to apprehend(catching pr arresting) any criminal or a person who violated the law.
- **Larceny**:- Taking of the personal property of any other person or business.
- **Arson**:- Arson means intentionally burning of other things like building, motor vehicles, forest or any property.
- **Rape**:- physical harassment given to any female.

- Link for the original Dataset: https://www.kaggle.com/sohier/crime-in-baltimore
- Link for the updated Dataset: https://drive.google.com/file/d/1t5rmHwOHd4zCgtVE0WVeN-PWtkT2VSkx/view
- Tool used for analysis is apache pig.

2.1 Sample Dataset

1,2017-09-02,1111-11-11 23:30:00,3JK,4200 AUDREY AVE,ROBBERY - RESIDENCE,I,KNIFE,913,SOUTHERN,Brooklyn,-76.60541,39.22951,ROW/TOWNHO,1

2,2017-09-02,1111-11-11 23:00:00,7A,800 NEWINGTON AVE,AUTO THEFT, O,, 133, CENTRAL, Reservoir Hill,-76.63217,39.3136,STREET,1

3,2017-09-02,1111-11-11 22:53:00,9S,600 RADNOR AV, SHOOTING, Outside,FIREARM,524,NORTHERN,Winston-Govans,-76.60697,39.34768,Street,1

4,2017-09-02,1111-11-11 22:50:00,4C,1800 RAMSAY ST,AGG. ASSAULT,I,OTHER,934, SOUTHERN, Carrollton Ridge,-76.64526,39.28315,ROW/TOWNHO,1

5,2017-09-02,1111-11-11 22:31:00,4E,100 LIGHT ST,COMMON ASSAULT, O, HANDS, 113,CENTRAL,Downtown West,-76.61365,39.28756,STREET,1

6,2017-09-02,1111-11-11 22:00:00,5A,CHERRYCREST RD, BURGLARY, I,, 922, SOUTHERN, Cherry Hill,-76.62131,39.24867,ROW/TOWNHO,1

7,2017-09-02,1111-11-11 21:15:00,1F,3400 HARMONY CT, HOMICIDE, Outside, FIREARM,232,SOUTHEASTERN,Canton,-76.56827,39.28202,Street,1

8,2017-09-02,1111-11-11 21:35:00,3B,400 W LANVALE ST,ROBBERY - STREET,O,,123,CENTRAL,Upton,-76.62789,39.30254,STREET,1

2.2 Analytical case study's on the dataset

Case study 1:- Top 5 places of most wanted district near which indoor crime takes place using knife from the analysis of the year 2015 to 2017.

Analysis report: Northeastern district of Maryland is most wanted area in case of criminal activity. In northeastern district Hillen, original northwood, stonewood-pentwood-winsto, Beverly hills and Rosemont east are the top 5 landmarks near which the highest indoor crimes are recorded between the year 2015-2017 using knife.

Analytical model for case study:

- fs -put baltimore.csv /
- a = load '/Baltimore.csv' using PigStorage(',') as (sno, date:datetime, time:datetime, code, address, description, io, weapon, post, district, nearest, longitude, latitude, premesis, total);
- b = group a by district;
- c = foreach b generate group, COUNT(a);
- d = order c by $1 DESC;

- e = limit d 1;
- f = filter a by district matches e.$0 and weapon matches 'KNIFE' and GetYear(date) <= 2017 and (io matches 'I' or io matches 'Inside');
- g = group f by nearest;
- h = foreach g generate group,COUNT(f);
- I = order h by $1;
- j = limit I 5;
- dump j;

case study 2:- The time duration of a day at winter season during which there should more security at NorthEastern district against crime.

Analysis report:- As NorthEastern district of Maryland is most wanted area in case of criminal activity, During winter season the maximum of crime cases are recorded in night. By the analysis of the dataset there should more security during the time period between 15:59:32 and 20:59:32.

Analytical model for case study:

- fs -put baltimore.csv /
- a = load '/Baltimore.csv' using PigStorage(',') as (sno, date:datetime, time:datetime, code, address, description, io, weapon, post, district, nearest, longitude, latitude, premesis, total);
- b = filter a by (GetMonth(date) >= 08 or GetMonth(date) <= 02) and disctrict matches 'NORTHEASTERN';
- c = group b by time;
- d = foreach c generate group,COUNT(b);
- e = order d by $1 DESC;
- f = limit e 5;
- x = order f by $0 DESC;
- g = limit x 1;
- h = order f by $0;
- I = limit h 1;
- j = join g by GetMonth($0),I by GetMonth($0);
- k = foreach j generate GetHour($0) as (h1:chararray), GetMinute($0) as (m1:chararray), GetSecond($0) as (s1:chararray), GetHour($2) as (h2:chararray), GetMinute($2) as (m2:chararray), GetSecond($2) as (s2:chararray);
- l = foreach k generate CONCAT('From ',$0,':',$1,':',$2,' To ',$3,':',$4,':',$5);
- dump l;

Case study 3:- The post, where less than crime cases were reported in the year 2013 and where no rape cases are reported.

Analysis report:- Although there are large number of criminal activity happening over Maryland but there are some post where less than 5 cases were reported in the year 2013 and those post are as follows. Post 115, 134, 135, 215, 216, 427, 516, 525, 535, 624, 625, 816 and 924 where one case is reported with no rape cases. Post 314, 725, 825, 834, 835 and 945 where two crime cases were reported with no rape cases. Post 217, 325, 436, 615 and 634 where total three crime cases were reported with no rape cases. Post 214 and 526 where four cases are reported with no rape cases.

Analytical model for case study:

- fs -put baltimore.csv /
- a = load '/Baltimore.csv' using PigStorage(',') as (sno, date:datetime, time:datetime, code, address, description, io, weapon, post, district, nearest, longitude, latitude, premesis, total);
- b = filter a by GetYear(date) == 2013;

- c = filter b by description matches 'RAPE';
- d = foreach c generate post as temppost;
- t = distincts d;
- e = join b by post LEFT OUTER,t by temppost;
- f = filter e by temppost is null;
- g = group f by post;
- h = foreach g generate group,COUNT(f);
- i = filter h by $1 < 5;
- dump i;

case study 4:- The hills throughout the city near which people loose there life normally due to the unwanted activity of other peoples like reckless driving of car on the road and murder for robbery. [suppose if more than 10 cases were reported in a particular place. It would be considered as dangerous place.]

Analysis report:- There are three hills near Baltimore city which are not so much safe for tourist because there are multiple accident cases had been reported due to unwanted activities of other people. Those hills are Cherry hill, Chipley hill and Reservoir hill.

Analytical model for case study:

- fs -put baltimore.csv /
- a = load '/Baltimore.csv' using PigStorage(',') as (sno, date:datetime, time:datetime, code, address, description, io, weapon, post, district, nearest, longitude, latitude, premesis, total);
- b = filter a by (nearest matches '.*Hill.*' or nearest matches '.*hill.*') and description matches 'HOMICIDE';
- c = group b by nearest;
- d = foreach c generate group,COUNT(b);
- e = filter d by $1 >= 10;
- dump e;

Case study 5:- Number of average attempt to crime are reported in a day in central and southern district for the year 2014 month wise.

Analysis report:- Throughout the world there are many cases when some dangerous crime activities are detected and stopped them before happening. Similarly there are some such cases near Baltimore city where attempt to crime are reported per day in central and southern district in 2014.

1) In the January month average 3 cases were reported per day.
2) In the February month average 4 cases were reported per day.
3) In the March month average 3 cases were reported per day.
4) In the April month average 3 cases were reported per day.
5) In the May month average 4 cases were reported per day.
6) In the June month average 4 cases were reported per day.
7) In the July month average 4 cases were reported per day.
8) In the August month average 4 cases were reported per day.
9) In the September month average 4 cases were reported per day.
10) In the October month average 5 cases were reported per day.
11) In the November month average 4 cases were reported per day.
12) In the December month average 4 cases were reported per day.

Analytical model for case study:

- fs -put baltimore.csv /
- a = load '/Baltimore.csv' using PigStorage(',') as (sno, date:datetime, time:datetime, code, address, description, io, weapon, post, district, nearest, longitude, latitude, premesis, total);
- b = filter a by GetYear(date) == 2014

and (district matches 'CENTRAL' or district matches 'SOUTHERN') and description matches 'BURGLARY';

- c = group b by date;
- d = foreach c generate group, COUNT(b);
- e = group d by GetMonth($0);
- f = foreach e generate group as (a:chararray),AVG(d.$1) as (b:chararray);
- g = foreach f generate CONCAT('In Month ',$0,' Average ',$1,' Attemp to crime cases are reported');
- dump g;

case study 6:- The post for each district where the security cops are most violent against the criminal and attempting the unlawful activities on them.

Analysis report:- There are many crime cases in the world where cops has to be strict towards criminal but, attempting unlawful actions on them is not correct. There is list of post for each district where cops are most violence with respect to other posts in the city.

1) In central district post 111 have 1824 such cases.
2) In eastern district post 331 have 740 such cases.
3) In western district post 123 have 780 such cases.
4) In northern district post 411 have 685 such cases.
5) In southern district post 922 have 913 such cases.
6) In northeastern district post 443 have 802 such cases.
7) In northwestern district post 614 have 585 such cases.
8) In southeastern district post 212 have 693 such cases.
9) In southwestern district post 842 have 602 such cases.

Analytical model for case study:

- fs -put baltimore.csv /
- a = load '/Baltimore.csv' using PigStorage(',') as (sno, date:datetime, time:datetime, code, address, description, io, weapon, post, district, nearest, longitude, latitude, premesis, total);
- b = filter a by description matches 'COMMON ASSAULT';
- c = group b by post;
- d = foreach c generate group, b.district, COUNT(b);
- e = foreach d generate $0,flatten($1),$2;
- f = distinct e;
- g = group f by $1;
- h = foreach g generate group, MAX(f.$2);
- i = join f by $1,h by $0;
- j = filter i by $2 == $4;
- h = foreach j generate $1,$0,$2;
- dump h;

Case study 7:- The safest hill near which tourist may stay.

Analysis report:- Crime and accident risk is everywhere in the world but, every people wants to stay at that place where the risk of crime and accident is less. Among all the hilly areas in the state of Maryland "Beverly hill" is the most safest place where a tourist may stay.

Analytical model for case study:

- fs -put baltimore.csv /
- a = load '/Baltimore.csv' using PigStorage(',') as (sno, date:datetime,

time:datetime, code, address, description, io, weapon, post, district, nearest, longitude, latitude, premesis, total);

- b = filter a by nearest matches '.*Hill.*' or nearest matches '.*hill.*';
- c = group b by nearest;
- d = foreach c generate group,COUNT(b);
- e = order d by $1;
- f = limit e 1;
- dump f;

3. CONCLUSION

In this review, I have examined different aspects and applications of Big Data Analytics. I have reviewed types of big data, its properties. I have also reviewed the tools and commands used in Big Data Analysis like Hadoop, HDFS Architecture and its commands, hadoop ecosystem, pig, data models of pig and its architecture.

Finally I have analyzed a dataset taken from kaggle.com by updating it according to my requirements. I have Analyzed some of Analytical case studies related to crime in city of Baltimore and its nearest places which may help the USA government to control criminal activities under city.

4. REFERENCES

1) https://intellipaat.com/blog/7-big-data-examples-application-of-big-data-in-real-life/
2) https://www.geeksforgeeks.org/hadoop-ecosystem/
3) https://intellipaat.com/blog/tutorial/hadoop-tutorial/big-data-overview/
4) https://www.guru99.com/what-is-big-data.html
5) https://www.edureka.co/blog/hdfs-commands-hadoop-shell-command
6) https://www.geeksforgeeks.org/hdfs-commands/
7) https://www.geeksforgeeks.org/introduction-to-apache-pig/
8) https://www.tutorialspoint.com/apache_pig/apache_pig_architecture.htm
9) https://www.kaggle.com/sohier/crime-in-baltimore
10) https://drive.google.com/file/d/1t5rmHwOHd4zCgtVE0WVeN-PWtkT2VSkx/view
11) https://www.researchgate.net/publication/264555968_Big_Data_Analytics_A_Literature_Review_Paper
12) https://www.nap.edu/read/23654/chapter/5