ANALYSIS OF CRIME BIG DATA USING MAPREDUCE

Kaushik Rajan
National College of Ireland
MSc Data Analytics
X17165849

Abstract: Crime has been on rise every year in the USA. In Washington DC, as per the statistics provided by the FBI, the amount of violent crimes that happen per 100,000 people is roughly about 1,330.2 and apart from this, 4,778.9 property crimes occur per 100,000. The government wants to take measure to decrease the crime rate and hence wants the statistics of the previous occurrences to gain insights on how to control the crimes. In this case study, the Big data containing the information on all the crime occurrences between the years 2008 - 2017 in Washington DC is investigated and analysed using Hadoop map reduce environments. The outputs gained from the map reduce operations are visually interpreted using R and Tableau for better understanding. The project addresses the queries such as year wise crimes committed, shift wise crimes committed and hour wise crimes committed in order to help the government to provide better security during the peak hours of crime occurrences.

Keywords: NoSQL, Hadoop, Map reduce, PIG, HBase, SQOOP

I. INTRODUCTION

Washington DC is one of the most unsafe state for people to live and is well know for all the criminal activities which is why it was nick named the murder capital in the early 1990’s [8]. The main reason for the increase in criminal activities is due to the rise in drug market. The government had taken various steps in order to eradicate the drug market in the 1990’s, as a result of which, the crime rate decreased in the early 2000’s but have been on rise again since then. In order to take measures and decrease the crime rates in Washington DC, the government must deep dive into the past data on various crimes happening around. To help the government, the Big data which contains the information on the crimes which happened between the years 2008 to 2017 is analysed. The dataset initially contains 31 various attributes, which is then decreased to 17 in order to process the data. The data cleaning part was done on RStudio. The factors that are taken into consideration in order to solve the queries of the government are Shift (Day, Evening, Midnight), Offence type, year, hour of the crime, crime-type, block and month. Then these attributes are used to find solve the novel queries such as – 1) In which shift has max number of crimes have occurred? 2) In which hour has max number of crimes happened? 3) Crime rate year wise. These queries help the government to spot the peak hour of the crime occurrences in order to set up special task force. The dataset is initially stored in MySQL after creating the schema, which is then moved onto the HDFS using SQOOP for further map reduce processing. The HDFS input is taken into Java eclipse to perform the map reduce function and the output is moved the NoSQL database – HBase. Further queries are done on PIG. The output obtained from the map reduce function and PIG query are visualized using R and Tableau for better understanding.

Section 2 focuses on the related work done by various researchers in the past. Section 3 contains information on the selected technologies. Section 4 covers the methodologies used in this project and section 5 and 6 covers the results obtained and the future work.

II. RELATED WORK

In the recent past, various techniques are being using in predicting where the next crime will take place, time – period during which the crime will take place, etc. Machine learning has been the most common way to do the tasks mentioned above. In 2006, SVM was used to find Crime Hot-Spots, i.e., to classify if a location is a crime hot spot or not. To do this, the researcher has used spatial dataset and use one – class SVM for predicting the hot-spot. In order to select the data, K-means clustering algorithm was used and selected portion of the data was labelled [1]. The researchers produced decent results by using the one-class SVM. Another research was made using similar kind of spatial databases and GIS. In this, various machine algorithms such as Decision Tree, Support vector machine (SVM), One Nearest Neighbour (INN), Naïve Bayes and Neural network with 2-layer network were used [2]. The overall performance of all the used methods were compared by using them for over a 10-month period, during which, they were compared based on the accuracy, precision, recall and F1. The results produced show that the performance of a complex algorithm is quite similar to the basic easy ones and INN performs better. In another research conducted, the researchers have used taxi flow data of the cities – Chicago and IL, and have used linear regression and negative binomial regression. The features used to do the regression were selected using feature selected technique and the results obtained from the regression models show that the features POI and taxi flow reduces the prediction error by 17.6% [3]. In 2017, models such as Z-CrimeTool, ID3 algorithm, hidden link detection, Naïve Bayes used in a project to predict the crimes and the end result shows that ID3 model performed better [4]. A similar analysis was to find the crime hotspot in Taiwan and for this analysis Big data containing the information on spatial data and drug related criminal activity. To do this, data mining classification methods such as random forest and Naïve Bayes were used and visualizations with the crime hotspots were plotted [5]. Naïve Bayes has been the most common method in classifying / predicting the crime hotspot so far to predict the Crime hotspot.

Rise in Big data has made it difficult to process the data, gain insights and to use algorithms and hence to process big data, Hadoop MapReduce environments are being preferred currently. In this project, we analyse the big data of crime using map reduce environments.

III. Chosen Technologies

MySQL: When it comes to an open source database management system, MySQL is the 1st choice as it can store such big data and can perform well. Querying is also simple and fast in MySQL and hence we have chosen the same to store the initial dataset.

SQOOP: SQOOP is used as the mediator as it is a tool designed to transfer big data fast and efficiently between Hadoop ecosystem and relational databases. We have used scoop to transfer the data since it can transfer data from the relational database to Hadoop framework with the same schema that was created in the relational database [6].

Java – Eclipse Environment: Eclipse is the most common IDE used by programmers and it consists of basic work space and plugins for customizing the environment. It consists of Hadoop plugin and has by default mapper and reducer libraries and uses Hadoop plugin to do the MapReduce function.

HBase: HBase is a NoSQL database which is used for storing Large datasets. It is written in Java and is developed on top of HDFS. HBase is column oriented database and it is similar to google big table. Main advantage of HBase is that it is fault tolerant, which means that data is not lost even if there is some issue with the database. HBase is available open source and hence HBase has been chosen to store the outputs from the MapReduce functions.

PIG: PIG's queries are simple and similar to MySQL queries and PIG performs well with all kinds of dataset and does the MapReduce functions well and is integrated with Hadoop framework [7].

RStudio: RStudio is an open source IDE for R and is most commonly used around the world as it is easy to programme in. It uses various packages to do the analysis job. RStudio is the choice for cleaning the dataset and for visualization.

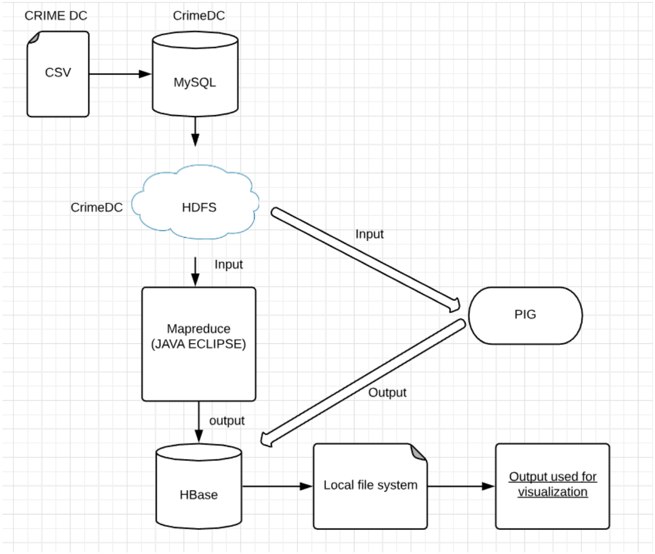
IV. Methodology

Description of the Dataset:

The dataset that was used for this project is taken from Kaggle - https://www.kaggle.com/vinchinzu/dc-metro-crime-data. It contains the information on the crimes in Washington DC and was last updated 1 year ago. The dataset contains the information for the years 2008 – 2017 and contains the following attributes (Only important attributes mentioned below).

Attribute name	Attribute information	Selected/Reason
SHIFT	Contains information on shift of the day (Day, Evening, Midnight)	Used to find shift wise crime
OFFENSE	Type of OFFENSE done (10 Factors)	Used to find count of each offence
year	year of crime (2008 – 2017)	Used to find year wise crime
hour	Hour during which crime happened (0 – 23)	Used to find hour wise crime
date	Date of the Crime	not used
BLOCK	Block where the crime happened	not used
day	Day in which the crime happened	not used
minute	Minute in which the crime happened	not used
second	Second in which the crime happened	not used
month	Month in which the crime happened	not used

Data Processing:



(Flowchart done on Lucidchart - https://www.lucidchart.com/documents/edit/3e8e4fe1-15fe-4f21-bc9c-d3ee6f4ae404/0)

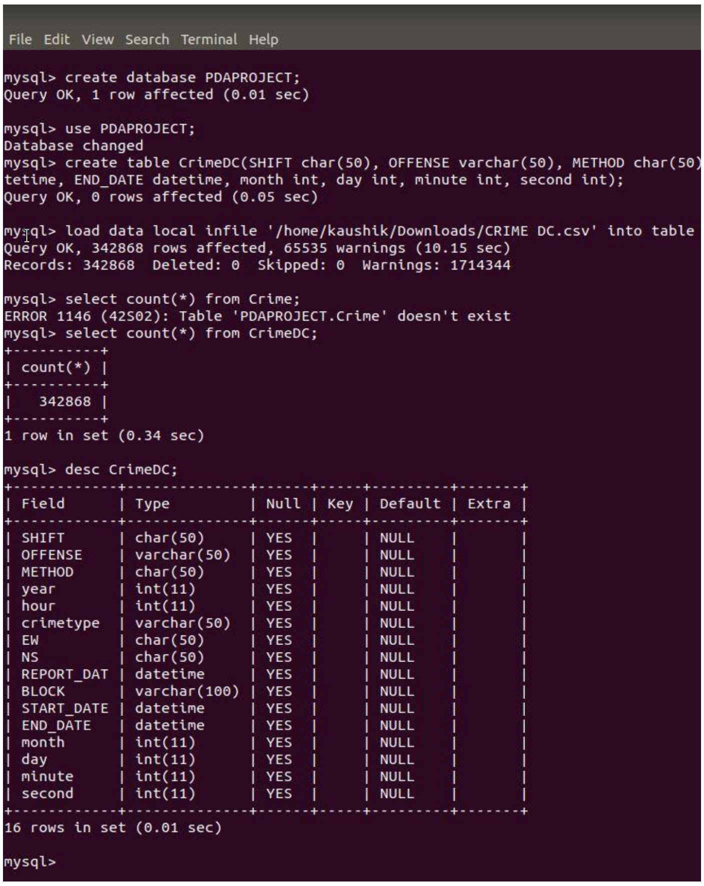
The Flowchart put up in the above image is the process that has been followed in this project.

Step 1: The dataset used for this project is initially downloaded by using '*sudo wget https://www.kaggle.com/vinchinzu/dc-metro-crime-data/downloads/dc-metro-crime-data.zip/5*' and is stored on local machine initially.

Step 2: The Dataset is then moved to MySQL after creating the database – PDA, table CrimeDC and finally the proper Schema of the table. The dataset is loaded into the created schema by using –

```
load data local infile
'/home/kaushik/Downloads/CRIME DC.csv' into table
CrimeDC fields terminated by ',' lines terminated by
'\n';
```

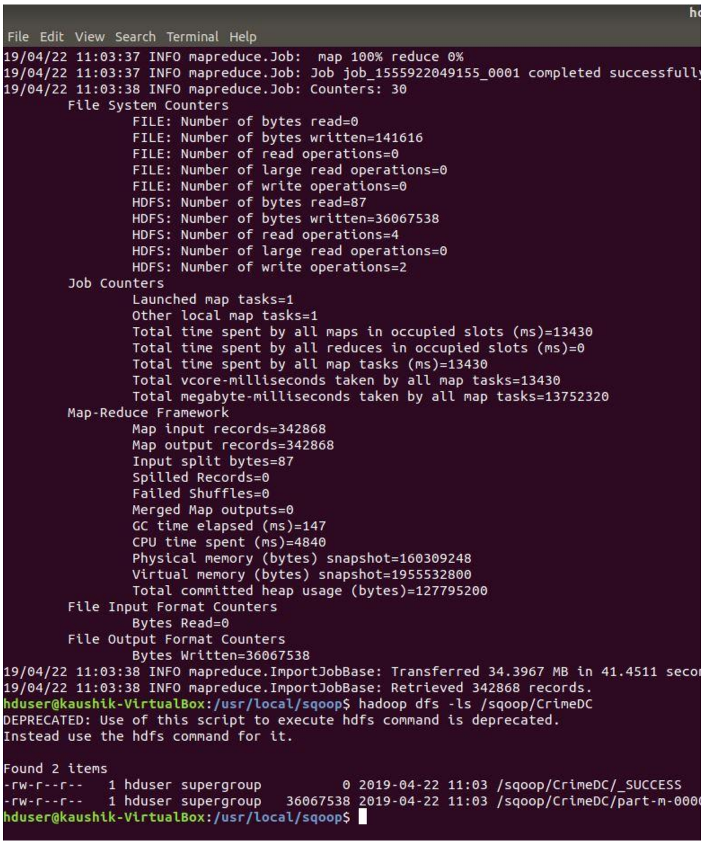
The image below shows the creation of schema, loading the table into MySQL, table count and description of the table.



```
mysql> create database PDAPROJECT;
Query OK, 1 row affected (0.01 sec)

mysql> use PDAPROJECT;
Database changed
mysql> create table CrimeDC(SHIFT char(50), OFFENSE varchar(50), METHOD char(50)
tetime, END_DATE datetime, month int, day int, minute int, second int);
Query OK, 0 rows affected (0.05 sec)

mysql> load data local infile '/home/kaushik/Downloads/CRIME DC.csv' into table
Query OK, 342868 rows affected, 65535 warnings (10.15 sec)
Records: 342868  Deleted: 0  Skipped: 0  Warnings: 1714344

mysql> select count(*) from Crime;
ERROR 1146 (42S02): Table 'PDAPROJECT.Crime' doesn't exist
mysql> select count(*) from CrimeDC;
+----------+
| count(*) |
+----------+
|   342868 |
+----------+
1 row in set (0.34 sec)

mysql> desc CrimeDC;
+------------+--------------+------+-----+---------+-------+
| Field      | Type         | Null | Key | Default | Extra |
+------------+--------------+------+-----+---------+-------+
| SHIFT      | char(50)     | YES  |     | NULL    |       |
| OFFENSE    | varchar(50)  | YES  |     | NULL    |       |
| METHOD     | char(50)     | YES  |     | NULL    |       |
| year       | int(11)      | YES  |     | NULL    |       |
| hour       | int(11)      | YES  |     | NULL    |       |
| crimetype  | varchar(50)  | YES  |     | NULL    |       |
| EW         | char(50)     | YES  |     | NULL    |       |
| NS         | char(50)     | YES  |     | NULL    |       |
| REPORT_DAT | datetime     | YES  |     | NULL    |       |
| BLOCK      | varchar(100) | YES  |     | NULL    |       |
| START_DATE | datetime     | YES  |     | NULL    |       |
| END_DATE   | datetime     | YES  |     | NULL    |       |
| month      | int(11)      | YES  |     | NULL    |       |
| day        | int(11)      | YES  |     | NULL    |       |
| minute     | int(11)      | YES  |     | NULL    |       |
| second     | int(11)      | YES  |     | NULL    |       |
+------------+--------------+------+-----+---------+-------+
16 rows in set (0.01 sec)

mysql>
```

Step 3: The table stored in the MySQL is moved to HDFS using SQOOP

'sqoop import --connect jdbc:mysql://127.0.0.1/PDA --username root --password Test1994 --table CrimeDC --target-dir /sqoop/CrimeDC -m 1'

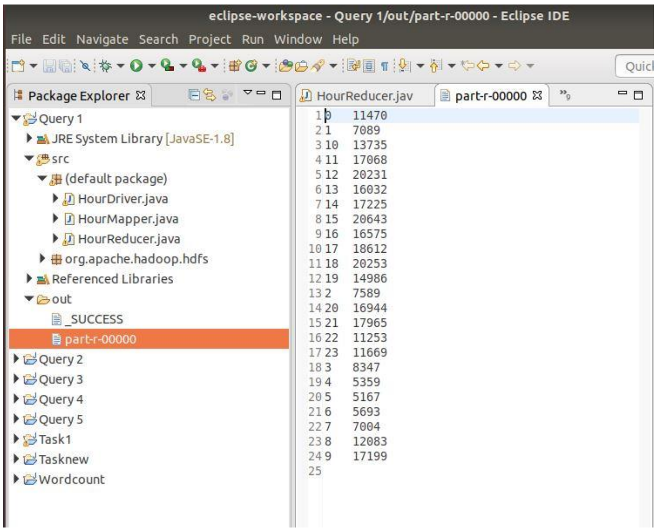


```
19/04/22 11:03:37 INFO mapreduce.Job:  map 100% reduce 0%
19/04/22 11:03:37 INFO mapreduce.Job: Job job_1555922049155_0001 completed successfully
19/04/22 11:03:38 INFO mapreduce.Job: Counters: 30
        File System Counters
                FILE: Number of bytes read=0
                FILE: Number of bytes written=141616
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=87
                HDFS: Number of bytes written=36067538
                HDFS: Number of read operations=4
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=2
        Job Counters
                Launched map tasks=1
                Other local map tasks=1
                Total time spent by all maps in occupied slots (ms)=13430
                Total time spent by all reduces in occupied slots (ms)=0
                Total time spent by all map tasks (ms)=13430
                Total vcore-milliseconds taken by all map tasks=13430
                Total megabyte-milliseconds taken by all map tasks=13752320
        Map-Reduce Framework
                Map input records=342868
                Map output records=342868
                Input split bytes=87
                Spilled Records=0
                Failed Shuffles=0
                Merged Map outputs=0
                GC time elapsed (ms)=147
                CPU time spent (ms)=4840
                Physical memory (bytes) snapshot=160309248
                Virtual memory (bytes) snapshot=1955532800
                Total committed heap usage (bytes)=127795200
        File Input Format Counters
                Bytes Read=0
        File Output Format Counters
                Bytes Written=36067538
19/04/22 11:03:38 INFO mapreduce.ImportJobBase: Transferred 34.3967 MB in 41.4511 secon
19/04/22 11:03:38 INFO mapreduce.ImportJobBase: Retrieved 342868 records.
hduser@kaushik-VirtualBox:/usr/local/sqoop$ hadoop dfs -ls /sqoop/CrimeDC
DEPRECATED: Use of this script to execute hdfs command is deprecated.
Instead use the hdfs command for it.

Found 2 items
-rw-r--r--   1 hduser supergroup          0 2019-04-22 11:03 /sqoop/CrimeDC/_SUCCESS
-rw-r--r--   1 hduser supergroup   36067538 2019-04-22 11:03 /sqoop/CrimeDC/part-m-0000
hduser@kaushik-VirtualBox:/usr/local/sqoop$
```

The above image shows that the table from MySQL has been moved to HDFS using SQOOP.

Step 4: MapReduce using Java Eclipse

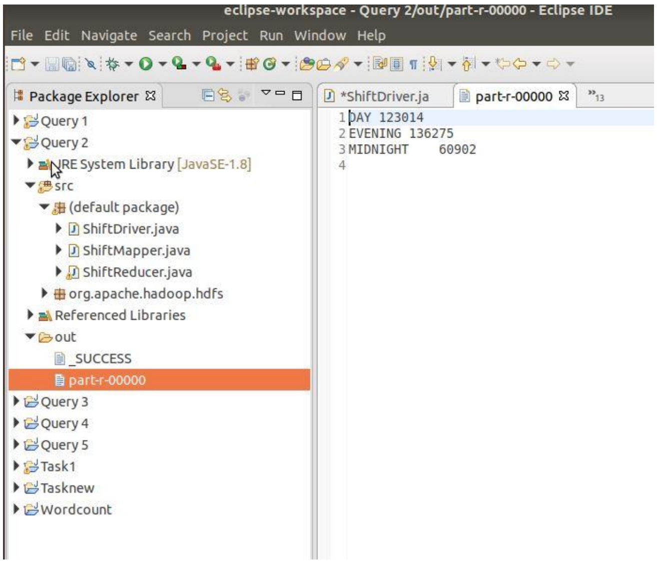
Map Reduce 1: Hour wise Crime Count



```
0	11470
1	7089
10	13735
11	17068
12	20231
13	16032
14	17225
15	20643
16	16575
17	18612
18	20253
19	14986
2	7589
20	16944
21	17965
22	11253
23	11669
3	8347
4	5359
5	5167
6	5693
7	7004
8	12083
9	17199
```

This MapReduce task is to find the number of crimes committed per hour division. The input table is taken from the HDFS and input path and the output path are set in the HourDriver. HourMapper maps the data based on the Hour variable and the reducer gives the sum of all the crimes hour wise.

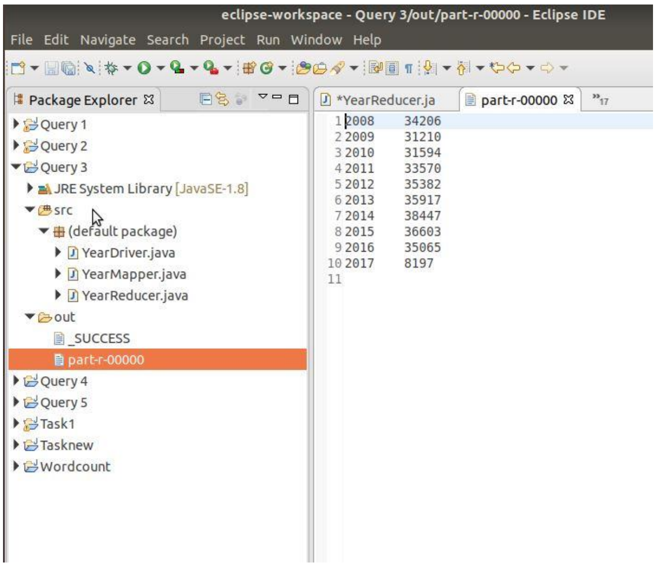
Map Reduce 2: Shift wise Crime Count



```
DAY 123014
EVENING 136275
MIDNIGHT     60902
```

This MapReduce task is to help the government to find during which shift the highest number of crimes occur which will be useful for increasing the security during that particular shift of the day.

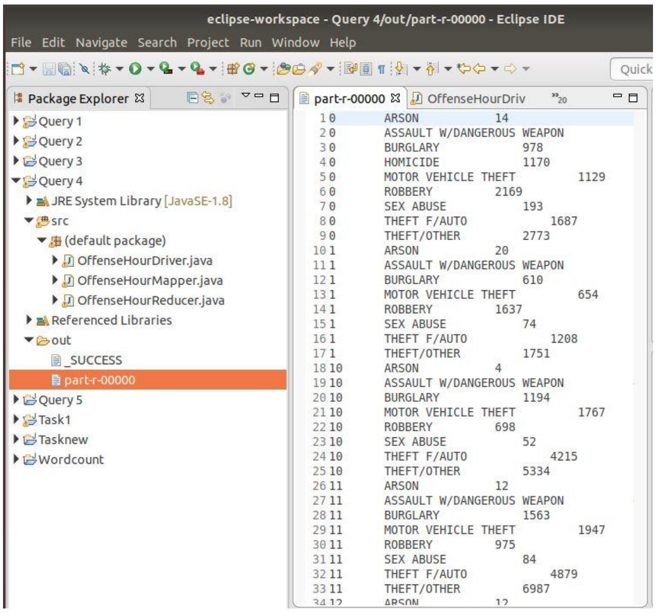
Map Reduce 3: Year wise Crime Count



```
2008	34206
2009	31210
2010	31594
2011	33570
2012	35382
2013	35917
2014	38447
2015	36603
2016	35065
2017	8197
```

In this case study, we compute the annual summary count. This is done to check if the annual count has decreased or increased throughout the years 2008 – 2017. The driver function gets the input from the HDFS input and sets the output path back to HDFS.

Map Reduce 4: Hour wise offence Count



```
0	ARSON		14
0	ASSAULT W/DANGEROUS WEAPON
0	BURGLARY	978
0	HOMICIDE	1170
0	MOTOR VEHICLE THEFT		1129
0	ROBBERY		2169
0	SEX ABUSE	193
0	THEFT F/AUTO		1687
0	THEFT/OTHER		2773
1	ARSON		20
1	ASSAULT W/DANGEROUS WEAPON
1	BURGLARY	610
1	MOTOR VEHICLE THEFT		654
1	ROBBERY		1637
1	SEX ABUSE	74
1	THEFT F/AUTO		1208
1	THEFT/OTHER		1751
10	ARSON		4
10	ASSAULT W/DANGEROUS WEAPON
10	BURGLARY	1194
10	MOTOR VEHICLE THEFT		1767
10	ROBBERY		698
10	SEX ABUSE	52
10	THEFT F/AUTO		4215
10	THEFT/OTHER		5334
11	ARSON		12
11	ASSAULT W/DANGEROUS WEAPON
11	BURGLARY	1563
11	MOTOR VEHICLE THEFT		1947
11	ROBBERY		975
11	SEX ABUSE	84
11	THEFT F/AUTO		4879
11	THEFT/OTHER		6987
12	ARSON		12
```

We do this case study to find out total sum of all the various offences which occurs each hour. It helps the government to find which type of occurs at which hour of the day.

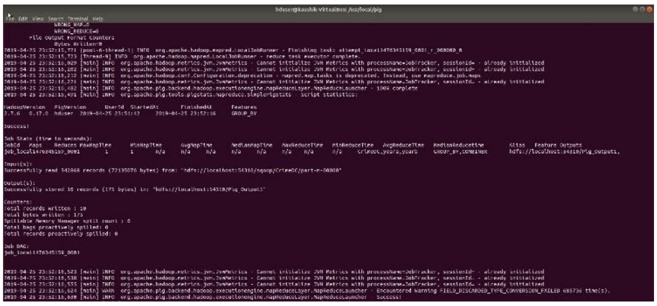
PIG QUERIES:

We have used 2 PIG queries to do the MapReduce operation.

PIG QUERY 1:

For this query, we have used the variable offence. The input is taken from HDFS which was loaded from MySQL to HDFS. We group by the OFFENSE variable and then take the count of it to find the count of various types of offence committed.

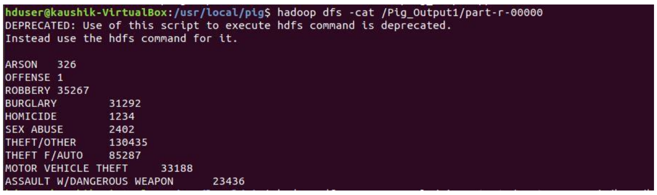
Below is the image of PIG running successfully.



```
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Output Format Counters
                Bytes Written=0
2019-04-25 23:52:15,721 [pool-6-thread-1] INFO  org.apache.hadoop.mapred.LocalJobRunner - Finishing task: attempt_local1476345159_0001_r_000000_0
2019-04-25 23:52:15,723 [Thread-9] INFO  org.apache.hadoop.mapred.LocalJobRunner - reduce task executor complete.
2019-04-25 23:52:16,029 [main] INFO  org.apache.hadoop.metrics.jvm.JvmMetrics - Cannot initialize JVM Metrics with processName=JobTracker, sessionId= - already initialized
2019-04-25 23:52:16,202 [main] INFO  org.apache.hadoop.metrics.jvm.JvmMetrics - Cannot initialize JVM Metrics with processName=JobTracker, sessionId= - already initialized
2019-04-25 23:52:16,212 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - mapred.map.tasks is deprecated. Instead, use mapreduce.job.maps
2019-04-25 23:52:16,221 [main] INFO  org.apache.hadoop.metrics.jvm.JvmMetrics - Cannot initialize JVM Metrics with processName=JobTracker, sessionId= - already initialized
2019-04-25 23:52:16,482 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - 100% complete
2019-04-25 23:52:16,491 [main] INFO  org.apache.pig.tools.pigstats.mapreduce.SimplePigStats - Script Statistics:

HadoopVersion   PigVersion      UserId  StartedAt       FinishedAt      Features
2.7.6   0.17.0  hduser  2019-04-25 23:51:42     2019-04-25 23:52:16     GROUP_BY

Success!

Job Stats (time in seconds):
JobId   Maps    Reduces MaxMapTime      MinMapTime      AvgMapTime      MedianMapTime   MaxReduceTime   MinReduceTime   AvgReduceTime   MedianReducetime        Alias   Feature Outputs
job_local1476345159_0001        1       1       n/a     n/a     n/a     n/a     n/a     n/a     n/a     crimedc,years,yearb     GROUP_BY,COMBINER       hdfs://localhost:54310/Pig_Output1,

Input(s):
Successfully read 342868 records (72135076 bytes) from: "hdfs://localhost:54310/sqoop/CrimeDC/part-m-00000"

Output(s):
Successfully stored 10 records (175 bytes) in: "hdfs://localhost:54310/Pig_Output1"

Counters:
Total records written : 10
Total bytes written : 175
Spillable Memory Manager spill count : 0
Total bags proactively spilled: 0
Total records proactively spilled: 0

Job DAG:
job_local1476345159_0001


2019-04-25 23:52:16,523 [main] INFO  org.apache.hadoop.metrics.jvm.JvmMetrics - Cannot initialize JVM Metrics with processName=JobTracker, sessionId= - already initialized
2019-04-25 23:52:16,538 [main] INFO  org.apache.hadoop.metrics.jvm.JvmMetrics - Cannot initialize JVM Metrics with processName=JobTracker, sessionId= - already initialized
2019-04-25 23:52:16,555 [main] INFO  org.apache.hadoop.metrics.jvm.JvmMetrics - Cannot initialize JVM Metrics with processName=JobTracker, sessionId= - already initialized
2019-04-25 23:52:16,624 [main] WARN  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - Encountered Warning FIELD_DISCARDED_TYPE_CONVERSION_FAILED 685736 time(s).
2019-04-25 23:52:16,680 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - Success!
```

hduser@kaushik-VirtualBox: /usr/local/pig
File Edit View Search Terminal Help

The output of the PIG script is stored in HDFS.

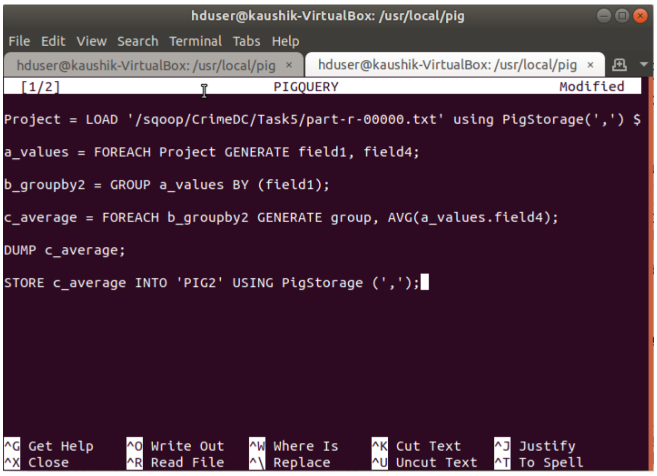


```
hduser@kaushik-VirtualBox:/usr/local/pig$ hadoop dfs -cat /Pig_Output1/part-r-00000
DEPRECATED: Use of this script to execute hdfs command is deprecated.
Instead use the hdfs command for it.

ARSON   326
OFFENSE 1
ROBBERY 35267
BURGLARY        31292
HOMICIDE        1234
SEX ABUSE       2402
THEFT/OTHER     130435
THEFT F/AUTO    85287
MOTOR VEHICLE THEFT     33188
ASSAULT W/DANGEROUS WEAPON      23436
```

PIG QUERY 2:

The second PIG query is done by taking the output of java eclipse MapReduce operation as input and finding the average of shift wise crime count.

Below is the PIG query which uses output taken from java which was stored in HDFS and does the average of the shift wise crime count.



```
 [1/2]                    PIGQUERY                           Modified

Project = LOAD '/sqoop/CrimeDC/Task5/part-r-00000.txt' using PigStorage(',') $

a_values = FOREACH Project GENERATE field1, field4;

b_groupby2 = GROUP a_values BY (field1);

c_average = FOREACH b_groupby2 GENERATE group, AVG(a_values.field4);

DUMP c_average;

STORE c_average INTO 'PIG2' USING PigStorage (',');
```

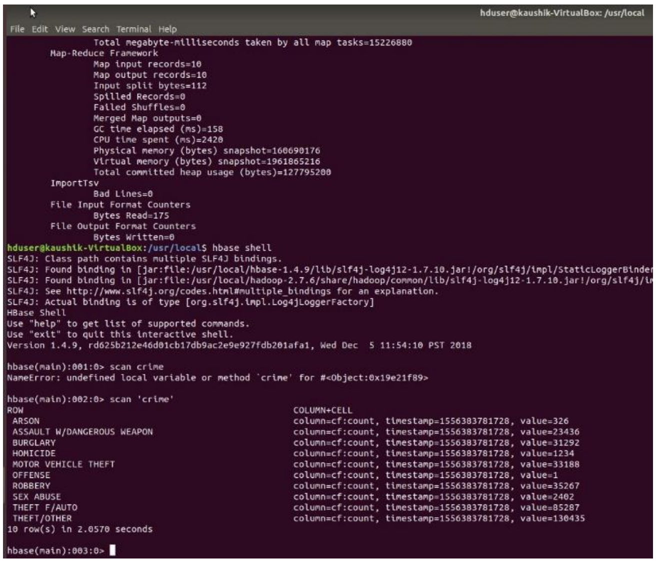
hduser@kaushik-VirtualBox: /usr/local/pig
File Edit View Search Terminal Tabs Help
hduser@kaushik-VirtualBox: /usr/local/pig
hduser@kaushik-VirtualBox: /usr/local/pig
^G Get Help
^X Close
^O Write Out
^R Read File
^W Where Is
^\ Replace
^K Cut Text
^U Uncut Text
^J Justify
^T To Spell

Step 5: Outputs of all the operations are moved to HBase which is a NoSQL database. HBase is capable of handling both unstructured and semi-structured data and is also fast while handling huge datasets.

The image below shows that the output of the MapReduce operation done on PIG is moved to HBase. Schema was initially created on HBase.

The output of the pig command is moved to Hbase using the following command.

hbase org.apache.hadoop.hbase.mapreduce.ImportTsv -Dimporttsv.columns=HBASE_ROW_KEY,cf:count crime '/Pig_Output1/part-r-00000'



hduser@kaushik-VirtualBox: /usr/local
File Edit View Search Terminal Help

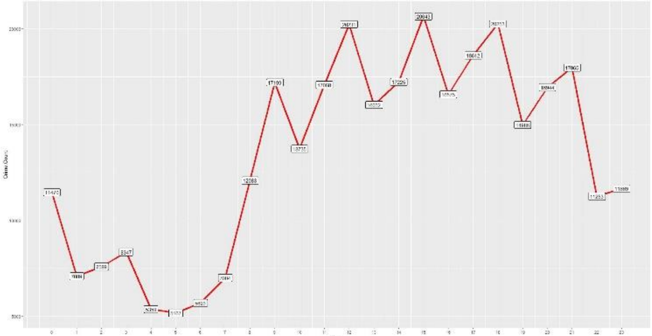
```
                Total megabyte-milliseconds taken by all map tasks=15226880
        Map-Reduce Framework
                Map input records=10
                Map output records=10
                Input split bytes=112
                Spilled Records=0
                Failed Shuffles=0
                Merged Map outputs=0
                GC time elapsed (ms)=158
                CPU time spent (ms)=2420
                Physical memory (bytes) snapshot=160690176
                Virtual memory (bytes) snapshot=1961865216
                Total committed heap usage (bytes)=127795200
        ImportTsv
                Bad Lines=0
        File Input Format Counters
                Bytes Read=175
        File Output Format Counters
                Bytes Written=0
hduser@kaushik-VirtualBox:/usr/local$ hbase shell
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/local/hbase-1.4.9/lib/slf4j-log4j12-1.7.10.jar!/org/slf4j/impl/StaticLoggerBinder
SLF4J: Found binding in [jar:file:/usr/local/hadoop-2.7.6/share/hadoop/common/lib/slf4j-log4j12-1.7.10.jar!/org/slf4j/in
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
HBase Shell
Use "help" to get list of supported commands.
Use "exit" to quit this interactive shell.
Version 1.4.9, rd625b212e46d01cb17db9ac2e9e927fdb201afa1, Wed Dec  5 11:54:10 PST 2018

hbase(main):001:0> scan crime
NameError: undefined local variable or method `crime' for #<Object:0x19e21f89>

hbase(main):002:0> scan 'crime'
ROW                                                 COLUMN+CELL
 ARSON                                              column=cf:count, timestamp=1556383781728, value=326
 ASSAULT W/DANGEROUS WEAPON                         column=cf:count, timestamp=1556383781728, value=23436
 BURGLARY                                           column=cf:count, timestamp=1556383781728, value=31292
 HOMICIDE                                           column=cf:count, timestamp=1556383781728, value=1234
 MOTOR VEHICLE THEFT                                column=cf:count, timestamp=1556383781728, value=33188
 OFFENSE                                            column=cf:count, timestamp=1556383781728, value=1
 ROBBERY                                            column=cf:count, timestamp=1556383781728, value=35267
 SEX ABUSE                                          column=cf:count, timestamp=1556383781728, value=2402
 THEFT F/AUTO                                       column=cf:count, timestamp=1556383781728, value=85287
 THEFT/OTHER                                        column=cf:count, timestamp=1556383781728, value=130435
10 row(s) in 2.0570 seconds

hbase(main):003:0> █
```

Step 6: The outputs are then taken out from HBase and moved to local memory. The outputs are then visualized using ggplot2 in RStudio and Tableau.

V. Results and Visualization

This section covers the results of each of the operation carried out using java eclipse and PIG. The outputs have been visualized using RStudio and Tableau

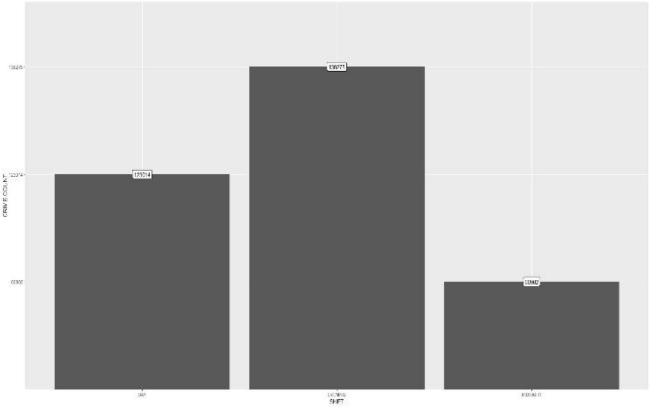
MapReduce 1



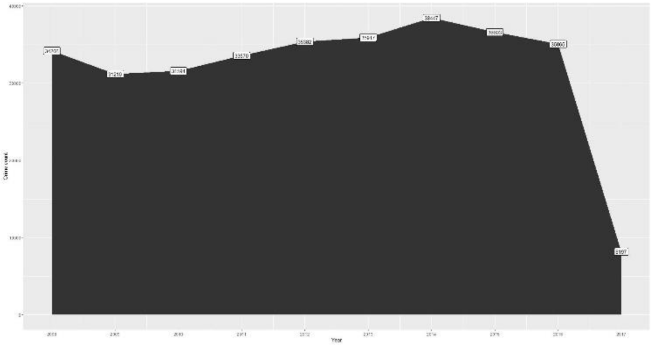
The output of MapReduce operation 1 is visualized using line graph on RStudio. The graph shows the hour-wise crime count and it can be inferred that the crime rate peaks at the 15th hour of the day with crime count of 20,643 and hence, the government can use this information to arrange better security during this hour of the day.

MapReduce 2

The output of the 2nd MapReduce task is to show the shift wise crime count which is a consolidation of task 1, which covers the hours of the day into different shifts. The choice of graph to represent this output is bar graph, which is plotted in RStudio. It can be inferred that, the highest number of crimes occur during the evening time with the crime count value of 1,36,275 which is said to be the time during when drug sales peak, which shows the correlation between the drug sales and crime count. Second highest number of crimes occur during day time with 1,23,014 crimes being record between the years 2008 – 2017.

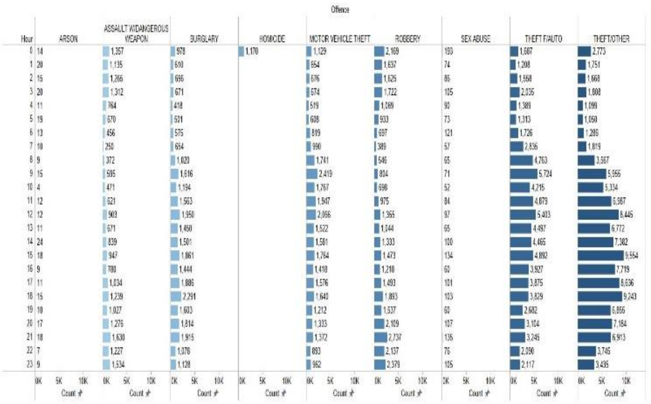


MapReduce 3



Area graph has been chosen to represent the year wise increase in crime rates and RStudio was the choice of visualization tool. The graph shows that there is a dip in crime rate during the years 2016 – 2017 but this is due to the fact that the data for 2017 was not updated with the info of crimes till the end of the year. Apart from this, we can see that the crime rate has been on constant increase from 2008-2016. The crime constantly rising and it reaches the peak in the year 2014 with crime count going up to 38,447.

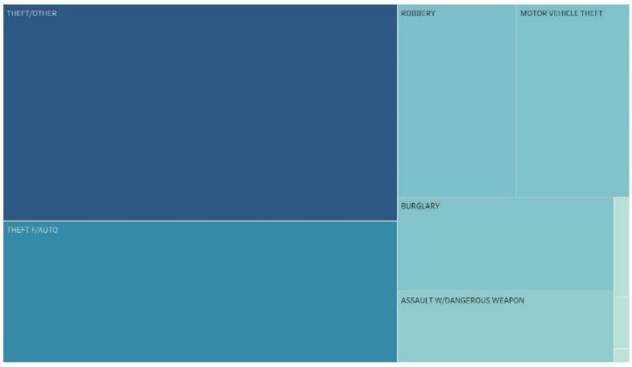
MapReduce 4



Side by side Bar graph was used to show the Hour wise offence count. The graph was done on Tableau. From the plot, we can see that theft has been the most common type of crime and it happens the most during the evening time which is the 15th hour with the crime count reaching to the value of 9,554.

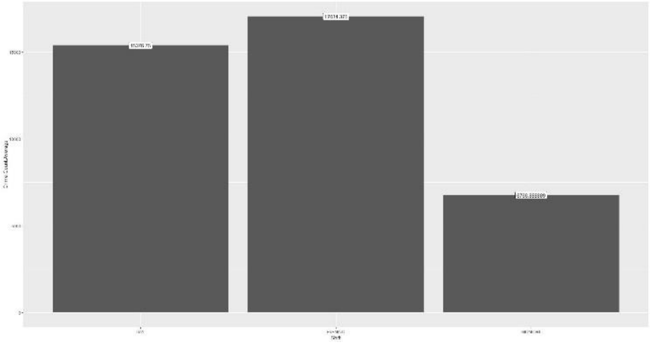
Pig 1

PIG Query is used to find the Crime Count by Offence. Tree map is used to show the offence wise crime count which was plotted using tableau. It can be inferred from the tree map that theft has been the most common mode of offence followed by robbery, motor vehicle theft, burglary and assault. Arson is the least mode of offence committed.



Pig 2

Pig query is used to find the average of shift wise crime count. Input for the query is taken from the MapReduce output. Bar graph is used to represent the output visually and the results show that on an average, more crimes occur during the evening time.



VI. Conclusion and Future work

The motive of this project was to analyse the Crime Big dataset to help the government set up better security to decrease the crime rate in Washington DC. We have used the technologies such as MySQL, HBase, Hadoop, Java, PIG, RStudio and Tableau for analysing the data. The Big data was analysed and visualized based on shift wise crime count, year wise crime count, hour wise crime count and offense wise crime count. The visualizations show that theft is the mostly committed crime, most crime occur during the evening shift and during the 15th hour. From the results, we can also understand that the crime rate has been constantly increasing from the years 2008 to 2017. By using this information, the government can act by increasing the security during the peak hour to decrease the crime rate and to make the punishment more severe for theft to decrease the theft count.

In the future, machine learning models can be done on pyspark to predict the crime hotspot and when and how will a crime happen in order to prevent the crime. Apart from that, apache spark can be used instead of Hadoop framework as it is said to be more efficient than Hadoop framework.

REFERENCES

[1] K. Kianmehr and R. Alhajj, "Crime Hot-Spots Prediction Using Support Vector Machine," *IEEE International Conference on Computer Systems and Applications, 2006.*, Dubai, UAE, 2006, pp. 952-959.
doi: 10.1109/AICCSA.2006.205203

keywords: {Support vector machines;Data mining;Support vector machine classification;Safety;Data analysis;Spatial databases;Cities and towns;Computer science;Machine learning;Geographic Information Systems},
URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1618468&isnumber=33913

[2] Yu, Jacky & Ward, Max & Morabito, Melissa & Ding, Wei. (2011). Crime Forecasting Using Data Mining Techniques. Proceedings - IEEE International Conference on Data Mining, ICDM. 779-786. 10.1109/ICDMW.2011.56.

[3] Wang, Hongjian & Kifer, Daniel & Graif, Corina & Li, Zhenhui. (2016). Crime Rate Inference with Big Data. 10.1145/2939672.2939736.

[4] C. Chauhan and S. Sehgal, "A review: Crime analysis using data mining techniques and algorithms," *2017 International Conference on Computing, Communication and Automation (ICCCA)*, Greater Noida, 2017, pp. 21-25.
doi: 10.1109/CCAA.2017.8229823
keywords: {data analysis;data mining;police data processing;data mining techniques;methodical approach;trends;increasing origin;computerized systems;crime data analysts;Law enforcement officers;unstructured data;predictive policing means;analytical techniques;predictive techniques;increased crime rate;criminals;crime analysis;pattern identification;pattern analysis;crime solving;advance technologies;Data mining;Tools;Algorithm design and analysis;Classification algorithms;Conferences;Forensics;Prediction algorithms;Data Mining;crime analysis;Naive Bayes Classifiers;Predictive approach},
URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8229823&isnumber=8229760

[5] Lin, Ying-Lung & Chen, Tenge-Yang & Yu, Liang-Chih. (2017). Using Machine Learning to Assist Crime Prevention. 1029-1030. 10.1109/IIAI-AAI.2017.46.

[6] https://sqoop.apache.org/

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

[8] https://en.wikipedia.org/wiki/Crime_in_Washington,_D.C.