

Analysis of Global Terrorism Database

For Edureka

By Roney Joseph

15th September 2014

Contents

Introduction.....	4
Database Columns.....	4
Dataset Volume	4
Detailed Analysis	4
Analysis Tools	5
Map Reduce Programming.....	5
Map Reduce Steps.....	8
Executing the Map Reduce Program (TerrorAnalysis.jar)	9
Map Reduce Output	10
Global Terrorism Incidences (1970 to 2013) Complete Output.....	11
Pig Programming	12
Pig script to Count Countrywise Terror Incidences.....	12
Executed the program using the following command:.....	12
Part of the MR output shown below, on executing the Pig script	13
Screen shot of the Part file (stored output file) on HDFS.....	14
Complete analysis report of Global Terror Count	14
Pig script to analyze and report highest terrorist incidence years (top 5 years)	20
Output of Stored Part File (top5years).....	20
Pig UDF Java Program (ISMAJOR).....	20
Pig Script.....	21
PIG UDF Execution.....	22
Part of the MR output shown below, on executing the Pig script	22
PIG UDF Output	23
Hive Programming.....	24

Project Documentation

Creating & Loading External Table GlobalTbl in Hive.....	24
Executing Hive Query Statements.....	24
Partitioned Table	26
Creating a Partitioned Table with Buckets.....	26
Setting the hive Partitioning shell parameters.....	26
Loading Data into Partitioned table	26
Partitioned Directories	27
Partitioned v/s Non Partitioned Data Analysis.....	28
Conclusions.....	28

Introduction

This document reflects the comprehensive analysis to be performed on the Global Terrorism Database (GTD), maintained by the National Consortium for the Study of Terrorism and Responses to Terrorism (START). The GTD is the result of a deliberative and consultative process that was initially undertaken by the [GTD Advisory Board](#) primarily between January and May 2006. Subsequent revisions to this database have been made by GTD staff, with the guidance of the Advisory Board.

It contains acts of terrorism from 1970 to the year 2013. GTD's goal is to be as transparent as possible regarding the data collection methodology, with a commitment to creating a highly comprehensive and consistent terrorism incident data set.

The analysis of GTD would provide critical data required to study the acts of terrorism based on the attacks perpetrated over the past 4 decades.

Database Columns

The dataset includes the GTD ID, incident date, incident location, incident information, attack information, target/victim information, perpetrator information, perpetrator statistics, weapon information, casualty information, consequences, additional information, and the data source information.

Dataset Volume

The Global Terrorism Database is a collection of over 1,50,000 terrorism incidents or data records.

Detailed Analysis

A detailed analysis is carried out on the comprehensive GTD to provide statistics on various categories.

It provides statistical information such as:-

- Year wise count of incidences (from 1970 to 2014)
- Country wise count of terrorism based incidences
- Major Indian cities & count of incidences
- Highest incidences years (top 5)
- Count of total number of casualties

- Highest Incidence based on Attack Types (ex: Assassination, Bombing etc)
- Count of attacks carried out by terror groups (top 5).

Analysis Tools

The dataset is loaded into HDFS & analyzed using Map Reduce jobs, Pig & Hive.

Map Reduce Programming

The objective of the TerrorMap program is to effectively use the backbone of the Hadoop environment, Map Reduce. TerrorMap program has 3 main classes, TerrorMap which extends Mapper, TerrorRed which extends Reducer and the driver class, TerrorDriver. The program processes all the Global terrorism records and generates the yearwise count of terror incidences.

Terror Map (Java program)

```
package org.ron;

import java.io.IOException;

import java.util.*;

import org.apache.hadoop.fs.Path;

import org.apache.hadoop.conf.*;

import org.apache.hadoop.io.*;

import org.apache.hadoop.util.*;

import org.apache.hadoop.mapreduce.Mapper;

import org.apache.hadoop.mapreduce.Reducer;

import org.apache.hadoop.conf.Configuration;

import org.apache.hadoop.conf.Configured;

import org.apache.hadoop.mapreduce.Job;

import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;

import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
```

Project Documentation

```
public class TerrorMap extends Mapper<LongWritable, Text, Text, IntWritable> {

    public final static IntWritable one = new IntWritable(1);

    @Override

    public void map(LongWritable key, Text value, Context context)

    throws IOException {

        String line = value.toString();

        String[] Incidence = line.split(",");

        String year = Incidence[1];

        try {

            context.write(new Text(year), one);

        } catch (InterruptedException e) {

            // TODO Auto-generated catch block

            e.printStackTrace();

        }

    }

}
```

Reducer Class (TerrorRed.java)

```
package org.ron;

import java.io.IOException;

import org.apache.hadoop.io.IntWritable;

import org.apache.hadoop.io.Text;

import org.apache.hadoop.mapreduce.Reducer;

public class TerrorRed extends Reducer<Text, IntWritable, Text, IntWritable>{
```

Project Documentation

@Override

public void reduce(Text key, Iterable<IntWritable> values, Context context) throws IOException, InterruptedException{

```
    int IncidenceFreq = 0;

    for (IntWritable value:values) {

        IncidenceFreq += value.get();

    }

    context.write(key, new IntWritable(IncidenceFreq));

}
```

Driver Class (TerrorDriver.java)

```
package org.ron;

import org.apache.hadoop.conf.Configuration;

import org.apache.hadoop.fs.Path;

import org.apache.hadoop.io.IntWritable;

import org.apache.hadoop.io.Text;

import org.apache.hadoop.mapreduce.Job;

import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;

import org.apache.hadoop.mapreduce.lib.input.TextInputFormat;

import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

import org.apache.hadoop.mapreduce.lib.output.TextOutputFormat;

public class TerrorDriver {

    public static void main(String[] args) throws Exception {

        Configuration conf = new Configuration();
```

Project Documentation

```
Job job = new Job(conf, "TerrorAnalysis");

job.setOutputKeyClass(Text.class);

job.setOutputValueClass(IntWritable.class);

job.setJarByClass(TerrorDriver.class);

job.setMapperClass(TerrorMap.class);

job.setReducerClass(TerrorRed.class);

job.setInputFormatClass(TextInputFormat.class);

job.setOutputFormatClass(TextOutputFormat.class);

FileInputFormat.addInputPath(job, new Path(args[0]));

FileOutputFormat.setOutputPath(job, new Path(args[1]));

System.out.println(job.waitForCompletion(true));

}

}
```

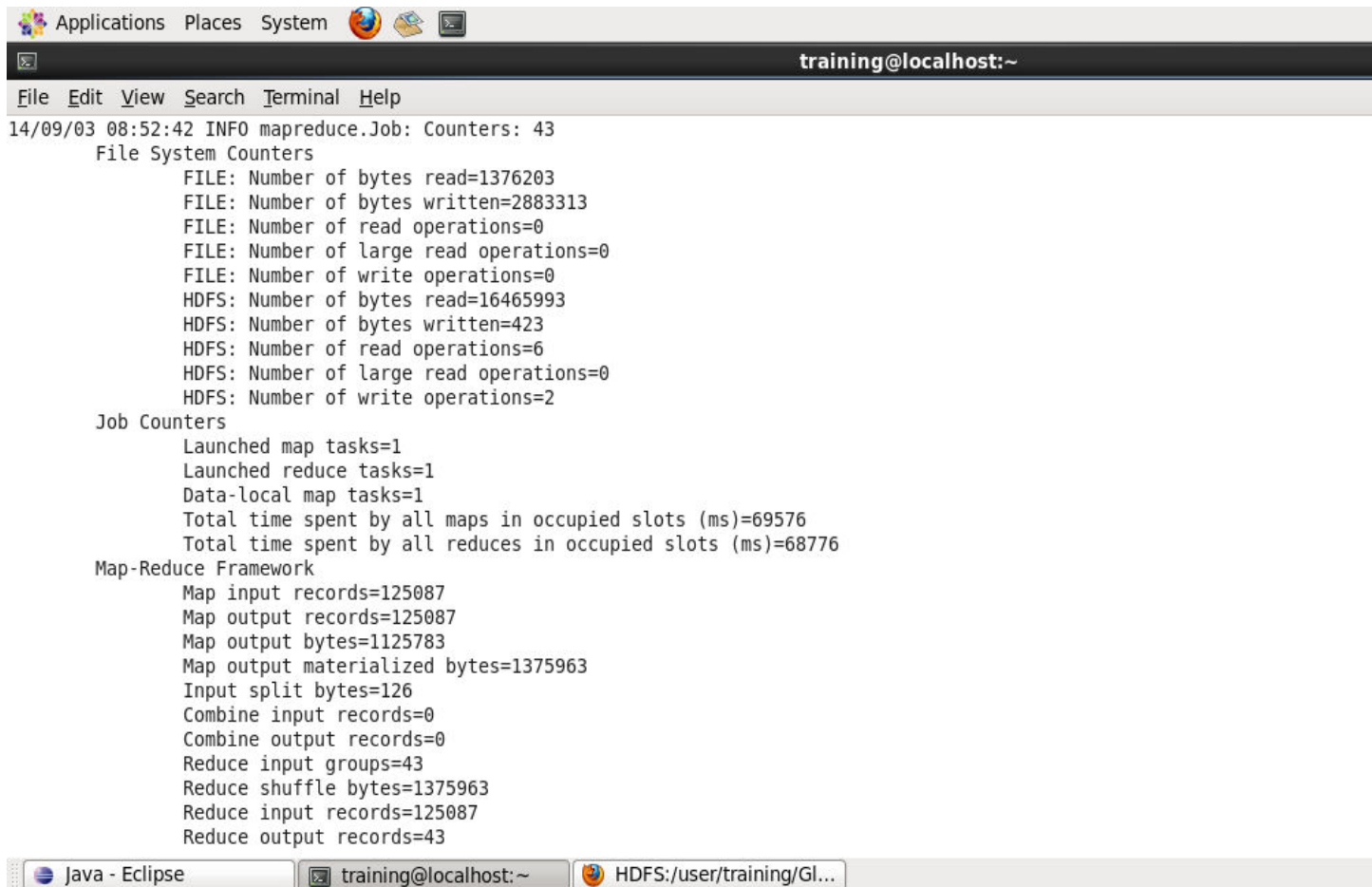
Map Reduce Steps

1. Coded the mapper, reducer and driver classes using eclipse
2. Imported the hadoop 2.0 libraries
3. Compiled and executed the code using Runas command
4. Exported the project as Runnable Jar to
/home/training/Desktop/programs/ExportedJavaFiles/TerrorAnalysis.jar
5. Transferred the Terrorismdb.csv dataset to local file system
6. Copied Terrorismdb.csv file to HDFS using the -put command
7. Executed the Map Reduce Program (TerrorAnalysis.jar)

Executing the Map Reduce Program (TerrorAnalysis.jar)

The map reduce program was executed (command below) that populated the GlobalTerrorOutput file on the HDFS.

```
hadoop jar/home/training/Desktop/programs/ExportedJavaFiles/TerrorAnalysis.jar
/user/training/ron/globalterrorismdb.csv /user/training/GlobalTerrorOutput
```

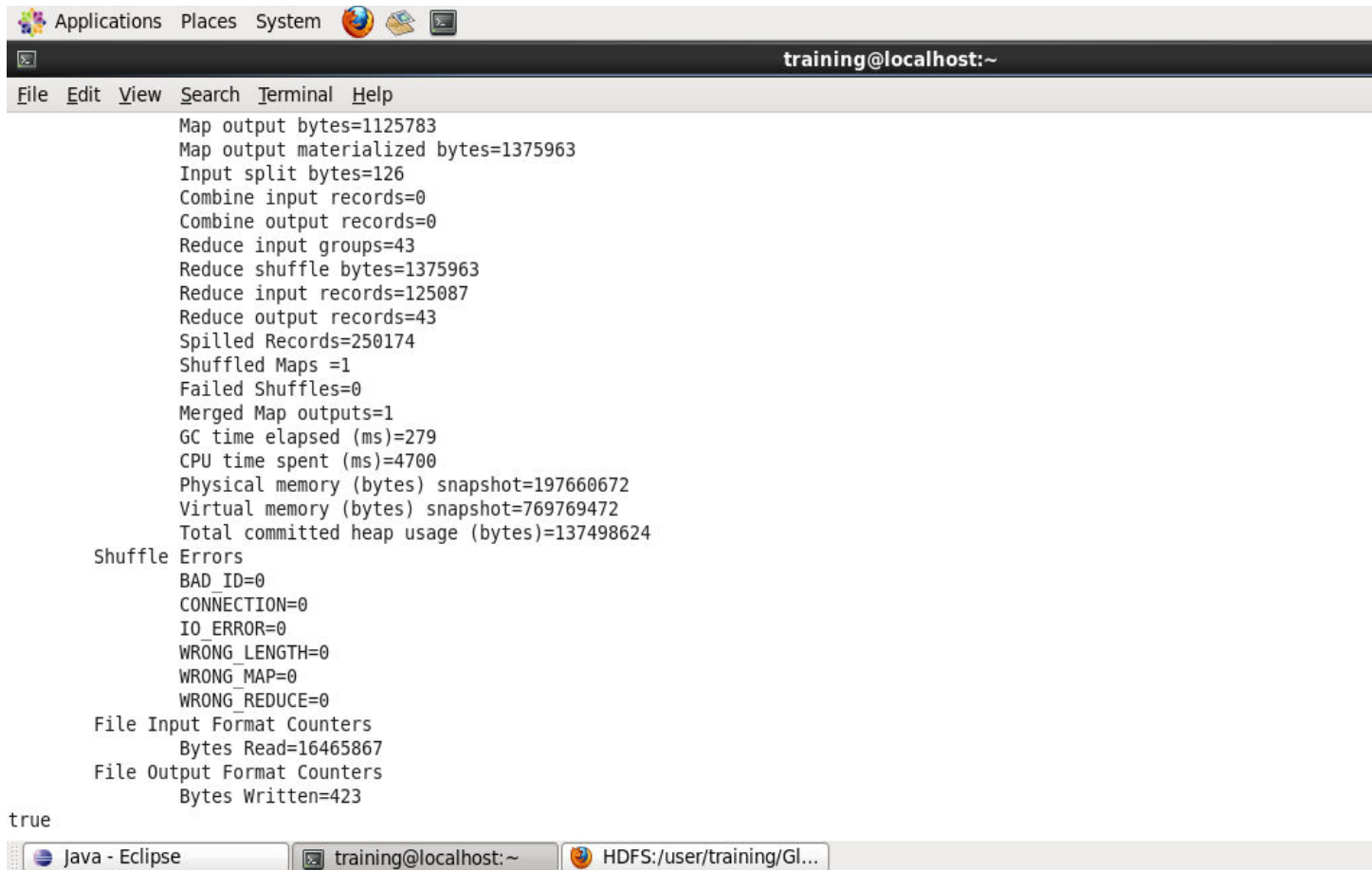


The screenshot shows a terminal window titled "training@localhost:~" within the Eclipse IDE. The terminal displays the output of a Hadoop MapReduce job. The output is structured into several sections: File System Counters, Job Counters, and Map-Reduce Framework. The File System Counters section shows the number of bytes read and written, as well as the number of read and write operations. The Job Counters section shows the number of launched map and reduce tasks, and the total time spent by all maps and reduces. The Map-Reduce Framework section shows the number of input and output records, bytes, and groups.

```
14/09/03 08:52:42 INFO mapreduce.Job: Counters: 43
  File System Counters
    FILE: Number of bytes read=1376203
    FILE: Number of bytes written=2883313
    FILE: Number of read operations=0
    FILE: Number of large read operations=0
    FILE: Number of write operations=0
    HDFS: Number of bytes read=16465993
    HDFS: Number of bytes written=423
    HDFS: Number of read operations=6
    HDFS: Number of large read operations=0
    HDFS: Number of write operations=2
  Job Counters
    Launched map tasks=1
    Launched reduce tasks=1
    Data-local map tasks=1
    Total time spent by all maps in occupied slots (ms)=69576
    Total time spent by all reduces in occupied slots (ms)=68776
  Map-Reduce Framework
    Map input records=125087
    Map output records=125087
    Map output bytes=1125783
    Map output materialized bytes=1375963
    Input split bytes=126
    Combine input records=0
    Combine output records=0
    Reduce input groups=43
    Reduce shuffle bytes=1375963
    Reduce input records=125087
    Reduce output records=43
```

The terminal window is part of the Eclipse IDE, with tabs for "Java - Eclipse", "training@localhost:~", and "HDFS:/user/training/Gl..." visible at the bottom.

Project Documentation



A screenshot of a terminal window titled "training@localhost:~". The window displays the output of a MapReduce job. The statistics are as follows:

```
Map output bytes=1125783
Map output materialized bytes=1375963
Input split bytes=126
Combine input records=0
Combine output records=0
Reduce input groups=43
Reduce shuffle bytes=1375963
Reduce input records=125087
Reduce output records=43
Spilled Records=250174
Shuffled Maps =1
Failed Shuffles=0
Merged Map outputs=1
GC time elapsed (ms)=279
CPU time spent (ms)=4700
Physical memory (bytes) snapshot=197660672
Virtual memory (bytes) snapshot=769769472
Total committed heap usage (bytes)=137498624

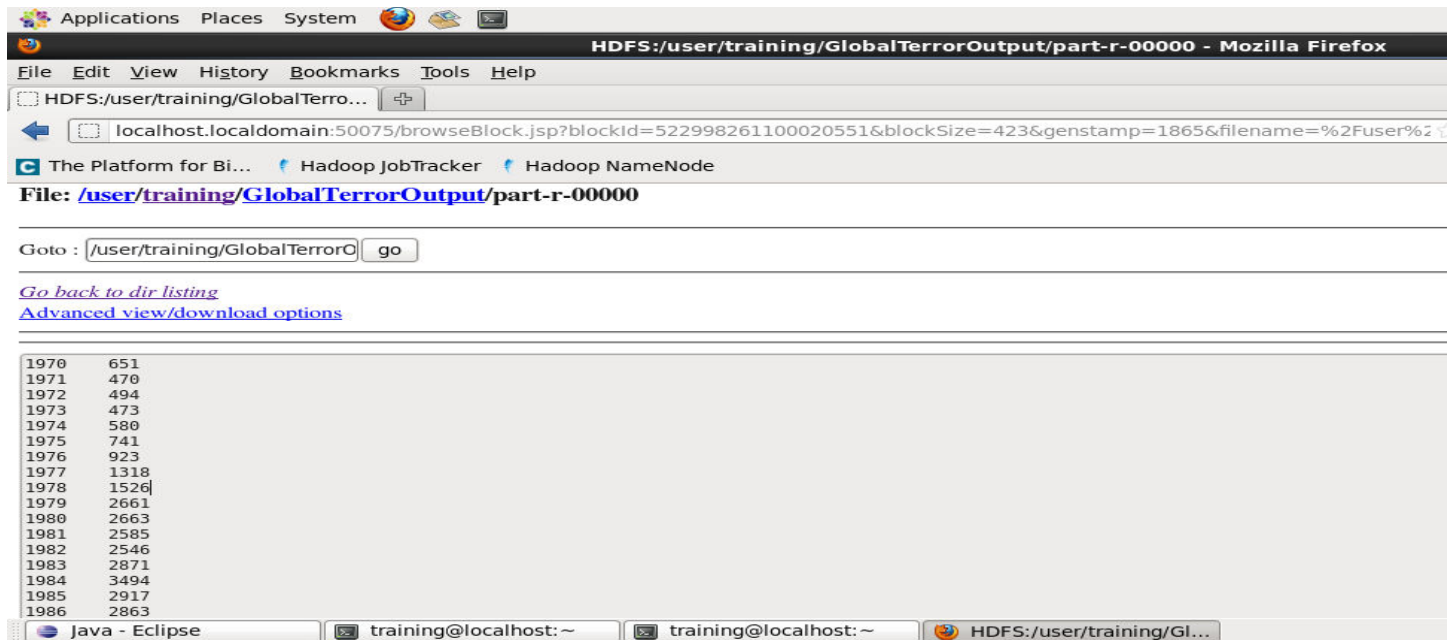
Shuffle Errors
BAD_ID=0
CONNECTION=0
IO_ERROR=0
WRONG_LENGTH=0
WRONG_MAP=0
WRONG_REDUCE=0

File Input Format Counters
Bytes Read=16465867
File Output Format Counters
Bytes Written=423

true
```

The terminal window has a menu bar with "File", "Edit", "View", "Search", "Terminal", and "Help". The status bar at the bottom shows "Java - Eclipse", "training@localhost:~", and "HDFS:/user/training/Gl..."

Map Reduce Output



A screenshot of a Mozilla Firefox browser window displaying the HDFS file details for `HDFS:/user/training/GlobalTerrorOutput/part-r-00000`. The browser shows the file's location, a "Goto" field with the path `/user/training/GlobalTerrorO`, and a "go" button. Below the "Goto" field, there are links for "Go back to dir listing" and "Advanced view/download options". The main content area displays a table of data points:

1970	651
1971	470
1972	494
1973	473
1974	580
1975	741
1976	923
1977	1318
1978	1520
1979	2661
1980	2663
1981	2585
1982	2546
1983	2871
1984	3494
1985	2917
1986	2863

The browser window has a menu bar with "File", "Edit", "View", "History", "Bookmarks", "Tools", and "Help". The status bar at the bottom shows "Java - Eclipse", "training@localhost:~", and "HDFS:/user/training/Gl..."

Global Terrorism Incidences (1970 to 2013) Complete Output

The list below provides year wise account of Terrorism incidences from 1970 to 2013 across the globe.

Year	Incidences
1970	651
1971	470
1972	494
1973	473
1974	580
1975	741
1976	923
1977	1318
1978	1526
1979	2661
1980	2663
1981	2585
1982	2546
1983	2871
1984	3494
1985	2917
1986	2863
1987	3186
1988	3721
1989	4322
1990	3887
1991	4684
1992	5078
1994	3459
1995	3083
1996	3058
1997	3204
1998	933
1999	1395
2000	1813
2001	1903
2002	1333
2003	1262
2004	1159
2005	2012
2006	2750
2007	3240

2008	4789
2009	4724
2010	4819
2011	5065
2012	8480
2013	11952

Pig Programming

Pig script to Count Countrywise Terror Incidences

--global terror count program

```
globalterror = load '/user/training/ron/globalterrorismdb' using PigStorage(',') as (eventid:int, iyear:int,
cntrycd:int, cntrynm:chararray, city:chararray, success:int, atktype:int, atktext:chararray, tgtype:int,
tgtypetxt:chararray, grpm:chararray, wpntype:int, wpntypetxt:chararray, nbrkill:int, dtasrc:chararray);
```

```
cntrygrp = GROUP globalterror by cntrynm;
```

```
terrorcount = FOREACH cntrygrp GENERATE group, COUNT(globalterror.cntrynm) AS cntrycnt;
```

```
outcount = ORDER terrorcount by cntrycnt DESC;
```

```
STORE outcount INTO '/user/training/globalterror/terrorcountout';
```

Executed the program using the following command:

```
$ pig terrorcount.pig
```

Part of the MR output shown below, on executing the Pig script

```

2014-09-18 02:28:19,919 [main] INFO org.apache.pig.tools.pigstats.SimplePigStats - Script Statistics:

HadoopVersion  PigVersion  UserId  StartedAt  FinishedAt  Features
2.0.0-cdh4.1.1  0.10.0-cdh4.1.1  training  2014-09-18 02:26:46  2014-09-18 02:28:19  GROUP_BY,ORDER_BY

Success!

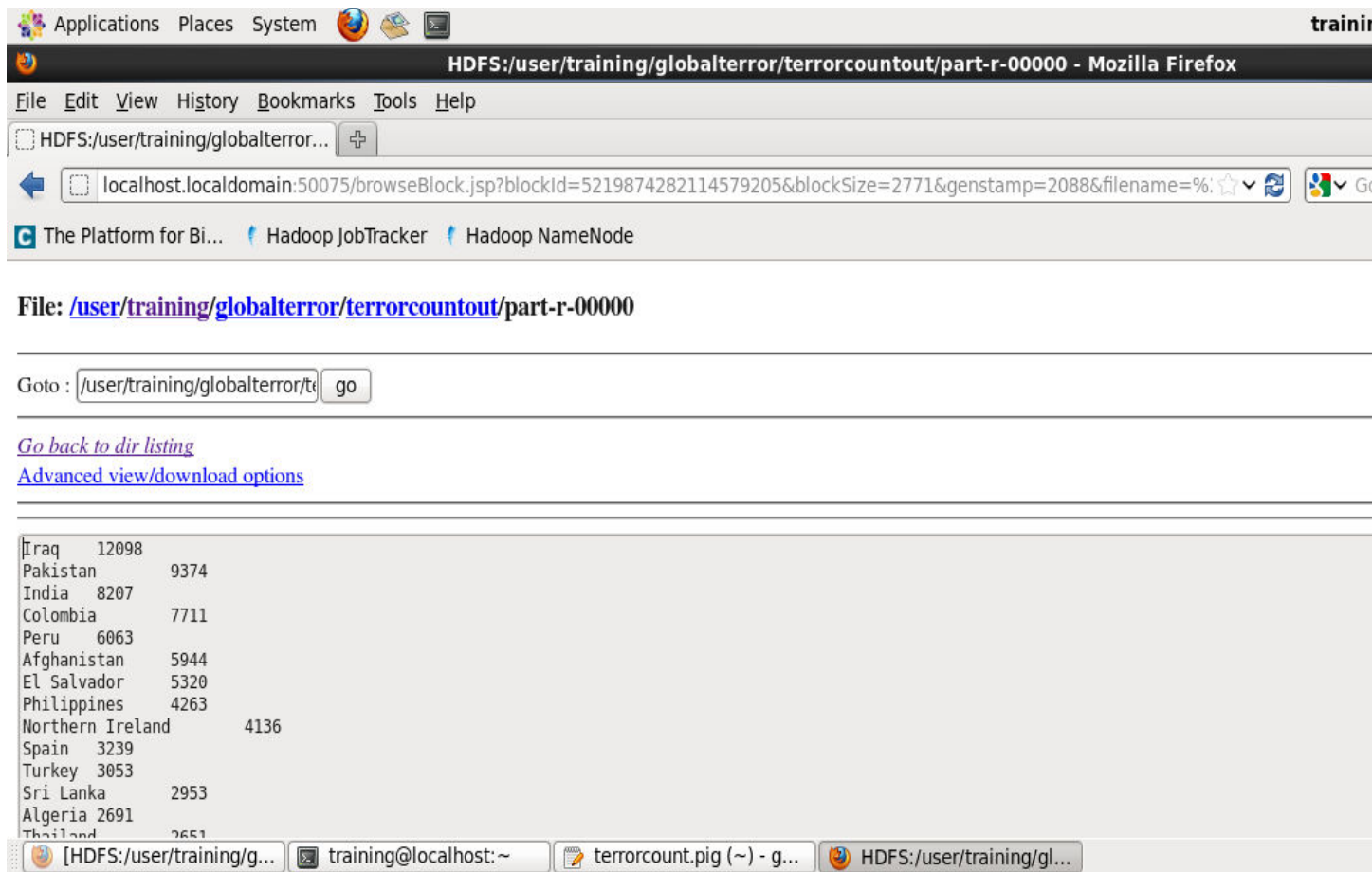
Job Stats (time in seconds):
JobId  Maps  Reduces  MaxMapTime  MinMapTime  AvgMapTime  MaxReduceTime  MinReduceTime  AvgReduceTime  Alias  Feature Outputs
job_1411020021741_0002  1  1  14  14  14  14  14  14  cntrygrp,globalterror,terrorcount  GROUP_BY,COMBINER
job_1411020021741_0003  1  1  5  5  5  5  5  5  5  outcount  SAMPLER
job_1411020021741_0004  1  1  5  5  5  5  5  5  5  outcount  ORDER_BY  /user/tr
aining/globalterror/terrorcountout,

Input(s):
Successfully read 125087 records (16519060 bytes) from: "/user/training/ron/globalterrorisymb"

Output(s):
Successfully stored 209 records (2771 bytes) in: "/user/training/globalterror/terrorcountout"

Counters:
Total records written : 209
Total bytes written : 2771
Spillable Memory Manager spill count : 0
Total bags proactively spilled: 0
Total records proactively spilled: 0
  
```

Screen shot of the Part file (stored output file) on HDFS



Applications Places System

HDFS:/user/training/globalterror/terrorcountout/part-r-00000 - Mozilla Firefox

File Edit View History Bookmarks Tools Help

HDFS:/user/training/globalterror/...

localhost.localdomain:50075/browseBlock.jsp?blockId=5219874282114579205&blockSize=2771&genstamp=2088&filename=%...

The Platform for Bi... Hadoop JobTracker Hadoop NameNode

File: [/user/training/globalterror/terrorcountout/part-r-00000](#)

Goto : go

[Go back to dir listing](#)

[Advanced view/download options](#)

Iraq	12098
Pakistan	9374
India	8207
Colombia	7711
Peru	6063
Afghanistan	5944
El Salvador	5320
Philippines	4263
Northern Ireland	4136
Spain	3239
Turkey	3053
Sri Lanka	2953
Algeria	2691
Thailand	2651

[HDFS:/user/training/g... training@localhost:~ terrorcount.pig (~) - g... HDFS:/user/training/gl...

Complete analysis report of Global Terror Count

Country	Incidence Count
Iraq	12098
Pakistan	9374
India	8207
Colombia	7711
Peru	6063
Afghanistan	5944
El Salvador	5320
Philippines	4263
Northern Ireland	4136
Spain	3239
Turkey	3053
Sri Lanka	2953
Algeria	2691

Project Documentation

Thailand	2651
United States	2381
Chile	2319
Lebanon	2170
Guatemala	2050
Russia	2032
Nicaragua	1968
South Africa	1949
Israel	1723
Somalia	1617
West Bank and Gaza Strip	1616
Nigeria	1535
Italy	1533
Corsica	1410
Yemen	1169
France	1154
Greece	1143
Bangladesh	916
Egypt	871
Nepal	869
Argentina	803
Indonesia	652
Iran	649
Syria	647
Great Britain	637
Germany	568
West Germany (FRG)	541
Angola	492
Mexico	475
Burundi	402
Japan	385
Sudan	371
Libya	369
Uganda	347
Kenya	331
Honduras	320
Bolivia	314
Myanmar	304
Venezuela	262
Brazil	259
Cambodia	258

Project Documentation

Congo (Kinshasa)	249
Puerto Rico	246
Mozambique	242
Ecuador	214
Haiti	212
Ireland	212
Georgia	209
Yugoslavia	205
China	188
Tajikistan	180
Kosovo	174
Rwanda	155
Bosnia-Herzegovina	150
Namibia	144
Ethiopia	141
Belgium	140
Portugal	139
Mali	131
Panama	126
Bahrain	123
Netherlands	122
Cyprus	122
Senegal	114
Macedonia	110
Switzerland	107
Austria	106
Zimbabwe	98
Sierra Leone	97
Dominican Republic	88
Jordan	86
Rhodesia	83
Papua New Guinea	81
Soviet Union	78
Australia	76
Saudi Arabia	76
Uruguay	75
Kuwait	73
Albania	71
Niger	69
Canada	68
Costa Rica	67

Project Documentation

Suriname	66
Sweden	61
Zambia	60
Malaysia	59
Central African Republic	58
Ivory Coast	56
Guadeloupe	56
Croatia	55
Ukraine	54
Tunisia	53
Paraguay	53
Chad	52
Togo	48
Bulgaria	48
Taiwan	47
Hungary	43
Azerbaijan	42
East Germany (GDR)	38
South Korea	36
Denmark	36
Morocco	35
Poland	34
Jamaica	33
Macau	33
Liberia	32
New Caledonia	31
Cameroon	31
Cuba	30
Hong Kong	26
Lesotho	25
Madagascar	25
Guyana	25
Tanzania	25
Kyrgyzstan	24
Kazakhstan	24
Congo (Brazzaville)	23
Guinea	22
Trinidad and Tobago	21
Uzbekistan	20
Moldova	20
United Arab Emirates	20

Project Documentation

Laos	20
Armenia	20
Malta	19
Djibouti	19
Norway	18
South Sudan	18
New Zealand	18
Mauritania	18
Czech Republic	18
Slovak Republic	17
Ghana	17
Fiji	17
Latvia	16
Luxembourg	16
Swaziland	15
Estonia	14
Belarus	13
Martinique	12
Serbia	11
Serbia-Montenegro	11
Eritrea	10
Vietnam	10
Czechoslovakia	10
Timor-Leste	10
Guinea-Bissau	9
Benin	8
Lithuania	8
Belize	8
Botswana	8
French Guiana	7
Singapore	7
Maldives	6
Slovenia	6
Finland	6
North Yemen	6
Qatar	6
Romania	6
Bahamas	5
Comoros	5
Western Sahara	5
Bhutan	5

Project Documentation

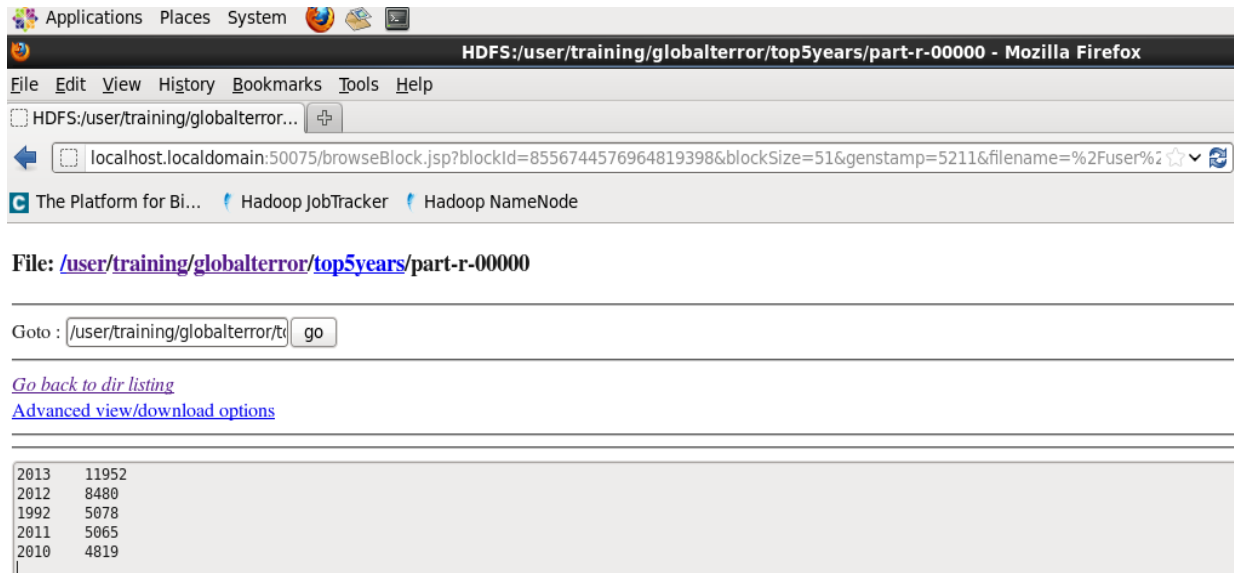
Grenada	5
Montenegro	4
Burkina Faso	4
Solomon Islands	4
Malawi	4
Gabon	4
Barbados	3
French Polynesia	3
Gambia	3
Iceland	3
Dominica	3
Equatorial Guinea	2
South Yemen	2
Seychelles	2
Antigua and Barbuda	2
Vanuatu	2
St. Kitts and Nevis	2
Mauritius	2
Gibraltar	1
Cayman Islands	1
Virgin Islands (U.S.)	1
South Vietnam	1
St. Lucia	1
Brunei	1
Vatican City	1
Turkmenistan	1
New Hebrides	1
International	1
Falkland Islands	1
Andorra	1
Wallis and Futuna	1
North Korea	1

Pig script to analyze and report highest terrorist incidence years (top 5 years)

The pig script below fetches the highest 5 years of terrorist attacks in the last 40 years.

```
--top 5 terror years
globalterror = load '/user/training/ron/globalterrorismdb' using PigStorage(',') as (eventid:int, iyear:int,
cntrycd:int, cntrynm:chararray, city:chararray, success:int, atktype:int, atktext:chararray, tgtype:int,
tgtypetxt:chararray, grpnm:chararray, wpntype:int, wpntypetxt:chararray, nbrkill:int, dtasrc:chararray);
grpyear = GROUP globalterror BY iyear;
grpyearcnt = FOREACH grpyear GENERATE group, COUNT(globalterror.iyear) AS YearCnt;
yearordcnt = ORDER grpyearcnt BY YearCnt DESC;
top5years = limit yearordcnt 5;
STORE top5years INTO '/user/training/globalterror/top5years';
```

Output of Stored Part File (top5years)



File: [/user/training/globalterror/top5years/part-r-00000](#)

Goto :

[Go back to dir listing](#)
[Advanced view/download options](#)

2013	11952
2012	8480
1992	5078
2011	5065
2010	4819

Pig UDF Java Program (ISMAJOR)

This program manages the ISMAJOR User Defined function in Pig by receiving the city as input and determines if the city is one of the Major Indian cities declared in this program. It returns the city name back to pig if its either Mumbai, Delhi, Bangalore, Chennai or Hyderabad.

Project Documentation

```
package org.ron.pigUDF;

import java.io.IOException;
import org.apache.pig.EvalFunc;
import org.apache.pig.data.Tuple;

public class ISMAJOR extends EvalFunc<String> {
    public String exec(Tuple input) throws IOException {
        String selcity;
        if (input == null || input.size() == 0)
            return null;
        try{
            String str = (String)input.get(0);
            if(str.equals("Mumbai") || str.equals("New Delhi") || str.equals("Bangalore") ||
                str.equals("Chennai") || str.equals("Hyderabad"))
            {
                selcity = str;
            }
            else {
                return null;
            }
            return selcity;
        }catch(Exception e){
            throw new IOException("Caught exception processing input row ", e);
        }
    }
}
```

Pig Script

```
--global terror count city UDF program
REGISTER /home/training/Desktop/programs/TerrorUDF.jar;
DEFINE ISMAJOR org.ron.pigUDF.ISMAJOR();
globalterror = load '/user/training/ron/globalterrorismdb' using PigStorage(',') as (eventid:int, iyear:int,
cntrycd:int, cntrynm:chararray, city:chararray, success:int, atktype:int, atktext:chararray, tgtype:int,
tgtypetxt:chararray, grpnm:chararray, wpntype:int, wpntypetxt:chararray, nbrkill:int, dtasrc:chararray);
FilterbyCnt = FILTER globalterror by cntrynm == 'India';
FilterbyCty = FOREACH FilterbyCnt GENERATE ISMAJOR(city) AS MajorCities;
FilteredMajorCities = FILTER FilterbyCty by MajorCities IS NOT NULL;

GroupMajorCities = GROUP FilteredMajorCities by MajorCities ;

CountMajorGroup = FOREACH GroupMajorCities GENERATE COUNT(FilteredMajorCities.MajorCities),
group as CitiesGroup;
Store CountMajorGroup INTO '/user/training/GlobalTerror/TerrorCityCount';
```

PIG UDF Execution

The Pig script containing the Pig UDF was executed from the command line. The UDF written in Java was called from within the pig script. The Major Indian cities predefined in the Java UDF program was used to select the tuples to process by pig. A group was created on the Major Cities and finally a count function was called to count the number of terrorism related acts in those cities from 1970 to 2013.

A Map Reduce job was automatically executed when the STORE command was encountered. The result was stored in a output file in the HDFS. The command to execute pig script, the job log and the final output is shown below:

```
$ pig TerrorcityUDF.pig
```

Part of the MR output shown below, on executing the Pig script

```

Success!

Job Stats (time in seconds):
JobId  Maps    Reduces MaxMapTime    MinMapTime    AvgMapTime    MaxReduceTime    MinReduceTime    AvgReduceTime    Alias F
eature Outputs
job_1411020021741_0005  1      1      8      8      8      7      7      7      CountMajorGroup,FilterbyCnt,FilterbyCty,
FilteredMajorCities,GroupMajorCities,globalterror  GROUP_BY,COMBINER  /user/training/GlobalTerror/TerrorCityCount,

Input(s):
Successfully read 125087 records (16519060 bytes) from: "/user/training/ron/globalterrorismdb"

Output(s):
Successfully stored 5 records (60 bytes) in: "/user/training/GlobalTerror/TerrorCityCount"

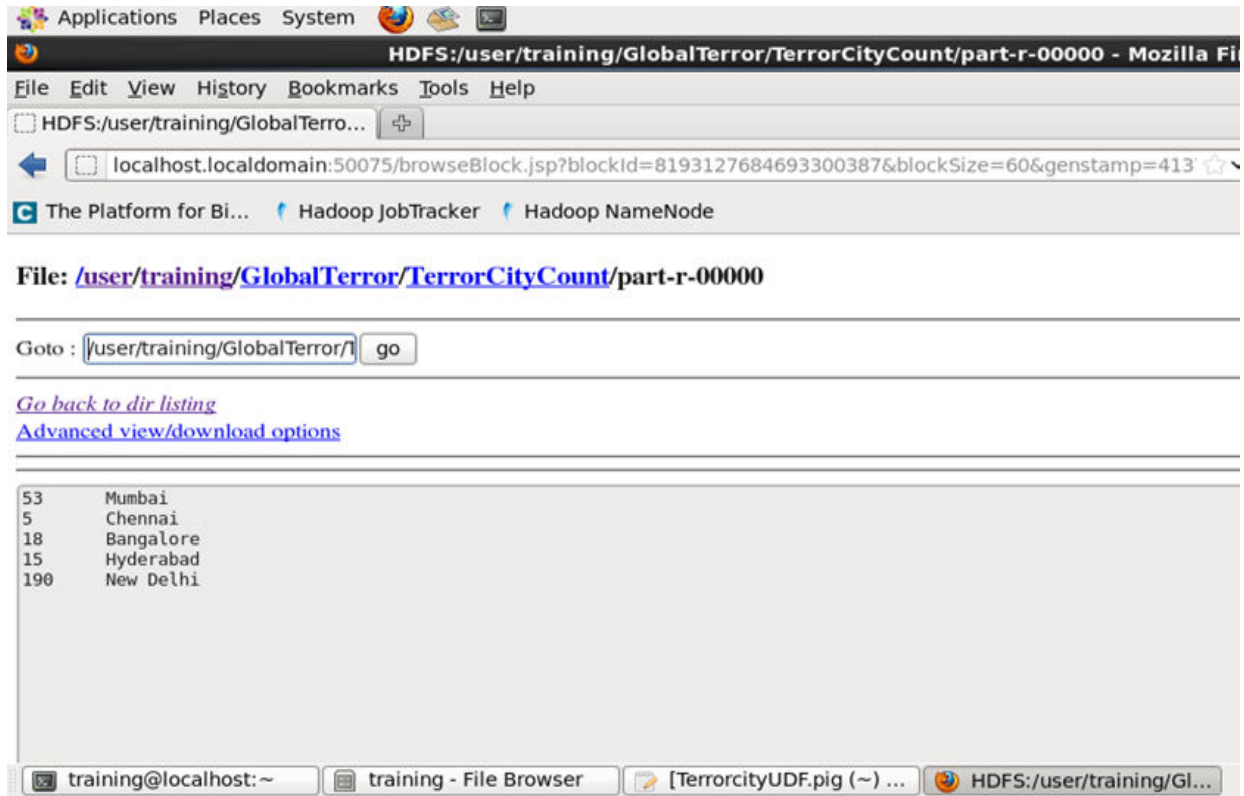
Counters:
Total records written : 5
Total bytes written : 60
Spillable Memory Manager spill count : 0
Total bags proactively spilled: 0
Total records proactively spilled: 0

Job DAG:
job_1411020021741_0005

2014-09-18 03:15:13,477 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - Success!
[training@localhost ~]$

```

PIG UDF Output



Applications Places System

HDFS:/user/training/GlobalTerror/TerrorCityCount/part-r-00000 - Mozilla Firefox

File Edit View History Bookmarks Tools Help

HDFS:/user/training/GlobalTerror/...

localhost.localdomain:50075/browseBlock.jsp?blockId=8193127684693300387&blockSize=60&genstamp=413'

The Platform for Bi... Hadoop JobTracker Hadoop NameNode

File: /user/training/GlobalTerror/TerrorCityCount/part-r-00000

Goto: /user/training/GlobalTerror/ go

[Go back to dir listing](#)

[Advanced view/download options](#)

53	Mumbai
5	Chennai
18	Bangalore
15	Hyderabad
190	New Delhi

training@localhost: ~ training - File Browser [TerrorcityUDF.pig (~) ... HDFS:/user/training/Gl...

Hive Programming

Creating & Loading External Table GlobalTbl in Hive

An external table was created and data loaded from a local file system.

```
hive> create EXTERNAL TABLE GlobalTbl(eventid STRING, iyear INT, cntrycd STRING, cntrynm STRING, city STRING, success INT, atkty
pe INT, atktext STRING, tgtype INT, tgtypetxt STRING, grpnm STRING, wpntype INT, wpntypetxt STRING, nbrkill INT, dtasrc STRING)
> ROW FORMAT DELIMITED
> FIELDS TERMINATED BY ','
> LINES TERMINATED BY '\n'
> STORED AS TEXTFILE
> LOCATION '/user/training/ron/GlobalTbl';
OK
Time taken: 0.069 seconds
hive> LOAD DATA LOCAL INPATH '/home/training/Desktop/globalterrorismdb.csv' INTO table GlobalTbl;
Copying data from file:/home/training/Desktop/globalterrorismdb.csv
Copying file: file:/home/training/Desktop/globalterrorismdb.csv
Loading data to table default.globaltbl
OK
Time taken: 1.793 seconds
```

Executing Hive Query Statements

Query 1. Fetching the count of people killed due to terrorism from 1970 to 2013 (available dataset):

```
hive> select sum(nbrkill) as TotalKilled from GlobalTbl;
Total MapReduce jobs = 1
Launching Job 1 out of 1
Number of reduce tasks determined at compile time: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapred.reduce.tasks=<number>
Starting Job = job_201409172050_0006, Tracking URL = http://localhost:50030/jobdetails.jsp?jobid=job_201409172050_0006
Kill Command = /usr/lib/hadoop/bin/hadoop job -Dmapred.job.tracker=localhost:8021 -kill job_201409172050_0006
2014-09-17 22:13:00,948 Stage-1 map = 0%, reduce = 0%
2014-09-17 22:13:03,966 Stage-1 map = 100%, reduce = 0%
2014-09-17 22:13:11,048 Stage-1 map = 100%, reduce = 100%
Ended Job = job_201409172050_0006
OK
258871
Time taken: 14.683 seconds
```


Query 2. Fetching the highest count of Terrorist Attack Types:

```
hive> SELECT atktext, COUNT(atktext) Cnt FROM globaltbl GROUP BY atktext order by Cnt DESC limit 1;
```

```
Kill Command = /usr/lib/hadoop/bin/hadoop job -Dmapred.job.tracker=localhost:8021 -kill  
job_201409152242_0009
```

```
2014-09-16 00:25:40,449 Stage-2 map = 0%, reduce = 0%
```

```
2014-09-16 00:25:42,461 Stage-2 map = 100%, reduce = 0%
```

```
2014-09-16 00:25:49,501 Stage-2 map = 100%, reduce = 33%
```

```
2014-09-16 00:25:50,506 Stage-2 map = 100%, reduce = 100%
```

```
Ended Job = job_201409152242_0009
```

OK

Bombing/Explosion 59424

Query 3 Listing the top 5 Terrorist Groups & their incidence counts

```
hive> select grpnm, COUNT(grpnm) as grpcnt from Globaltbl GROUP BY grpnm order by grpcnt DESC  
limit 5;
```

```
Starting Job = job_201409180708_0007, Tracking URL = http://localhost:50030/jobd  
etails.jsp?jobid=job_201409180708_0007  
Kill Command = /usr/lib/hadoop/bin/hadoop job -Dmapred.job.tracker=localhost:80  
21 -kill job_201409180708_0007  
2014-09-18 08:20:57,988 Stage-2 map = 0%, reduce = 0%  
2014-09-18 08:21:00,005 Stage-2 map = 100%, reduce = 0%  
2014-09-18 08:21:07,063 Stage-2 map = 100%, reduce = 100%  
Ended Job = job_201409180708_0007  
OK  
Unknown 56831  
Shining Path (SL) 4482  
Farabundo Marti National Liberation Front (FMLN) 3349  
Taliban 3204  
Irish Republican Army (IRA) 2633  
Time taken: 28.201 seconds  
hive>
```

Partitioned Table

A partitioned table was created with partition on country name which was split into 20 buckets on incidence year. The partitions and buckets ensures efficient pruning of data (better performance)when queries are executed on country and year.

Creating a Partitioned Table with Buckets

```
create TABLE GlobalPartTbl(eventid STRING, iyear INT, cntrycd STRING, city STRING, success INT, atktype INT, atktext STRING, tgtype INT, tgtypetxt STRING, grpnm STRING, wpntype INT, wpntypetxt STRING, nbrkill INT, dtasrc STRING)
```

```
> PARTITIONED BY (cntrynm STRING)
> CLUSTERED BY (iyear) INTO 20 BUCKETS
> ROW FORMAT DELIMITED
> FIELDS TERMINATED BY ','
> LINES TERMINATED BY '\n'
> STORED AS TEXTFILE;
```

Setting the hive Partitioning shell parameters

```
set hive.exec.dynamic.partition.mode = nonstrict;
set hive.exec.dynamic.partition = true;
SET hive.exec.max.dynamic.partitions.pernode=100000;
SET hive.exec.max.dynamic.partitions=100000;
```

Loading Data into Partitioned table

```
FROM GlobalTbl gtb INSERT OVERWRITE TABLE GlobalPartTbl PARTITION(cntrynm)

Select gtb.eventid, gtb.iyear, gtb.cntrycd, gtb.city, gtb.success, gtb.atktype, gtb.atktext,
gtb.tgtype, gtb.tgtypetxt, gtb.grpnm, gtb.wpntype, gtb.wpntypetxt, gtb.nbrkill,
gtb.dtasrc, gtb.cntrynm

DISTRIBUTE BY cntrynm;
```

Partitioned Directories

File Edit View History Bookmarks Tools Help

http://localhost.localdomain:50075/browseDirectory.jsp?dir=%2Fuser%2Fhive%2Fwa... Google

HDFS:/user/hive/warehouse/gt.d...

Contents of directory [/user/hive/warehouse/gt.db/globalparttbl/cntrynm=India](#)

Goto :

[Go to parent directory](#)

Name	Type	Size	Replication	Block Size	Modification Time	Permission	Owner	Group
000000_0	file	22.79 KB	1	64 MB	2014-09-17 11:56	rw-r--r--	training	supergroup
000001_0	file	28.25 KB	1	64 MB	2014-09-17 11:56	rw-r--r--	training	supergroup
000002_0	file	21.97 KB	1	64 MB	2014-09-17 11:56	rw-r--r--	training	supergroup
000003_0	file	27.19 KB	1	64 MB	2014-09-17 11:56	rw-r--r--	training	supergroup
000004_0	file	29.32 KB	1	64 MB	2014-09-17 11:56	rw-r--r--	training	supergroup

HDFS:/user/hive/warehouse/gt.d...

File: [/user/hive/warehouse/gt.db/globalparttbl/cntrynm=India/000000_0](#)

Goto :

[Go back to dir listing](#)

[Advanced view/download options](#)

```
200003040002,2000,92,Kachnara,1,1,Assassination,4,Military,People's War Group
(PWG),5,Firearms,13,CETIS
200003210002,2000,92,Chadisinghpooora,1,2,Armed Assault,14,Private Citizens & Property,Lashkar-
e-Taiba (LeT),5,Firearms,35,CETIS
200009240005,2000,92,Rajouri,1,3,Bombing/Explosion,4,Military,Unknown,6,Explosives/Bombs
/Dynamite,0,CETIS
200008170004,2000,92,Kotdhara,1,2,Armed Assault,14,Private Citizens & Property,Muslim
Militants,5,Firearms,6,UMD Schmid 2012
200008190001,2000,92,Indeh,1,2,Armed Assault,14,Private Citizens & Property,Muslim
Militants,5,Firearms,4,UMD Schmid 2012
200004090004,2000,92,Unknown,1,6,Hostage Taking (Kidnapping),2,Government (General),National
Liberation Front of Tripura (NLFT),13,Unknown,0,UMD Schmid 2012
```

Partitioned v/s Non Partitioned Data Analysis

The screen below shows the performance benefit of a Partitioned table over a non partitioned table by executing the same query on both types of tables:

```
Kill Command = /usr/lib/hadoop/bin/hadoop job -Dmapred.job.tracker=localhost:8021 -kill job_201409172050_0009
2014-09-17 22:34:06,648 Stage-1 map = 0%, reduce = 0%
2014-09-17 22:34:08,661 Stage-1 map = 100%, reduce = 0%
2014-09-17 22:34:16,723 Stage-1 map = 100%, reduce = 100%
Ended Job = job_201409172050_0009
OK
8207
Time taken: 13.644 seconds
select count(*) from globalparttbl where cntrynm = 'India';
Total MapReduce jobs = 1
Launching Job 1 out of 1
Number of reduce tasks determined at compile time: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapred.reduce.tasks=<number>
Starting Job = job_201409172050_0010, Tracking URL = http://localhost:50030/jobdetails.jsp?jobid=job_201409172050_0010
Kill Command = /usr/lib/hadoop/bin/hadoop job -Dmapred.job.tracker=localhost:8021 -kill job_201409172050_0010
2014-09-17 22:35:42,670 Stage-1 map = 0%, reduce = 0%
2014-09-17 22:35:44,683 Stage-1 map = 0%, reduce = 100%
Ended Job = job_201409172050_0010
OK
0
Time taken: 5.56 seconds
```

Conclusions

The GTD analysis project, has provided great insight on terrorism, which is impacting every country around the globe for the past 4 decades. The project has thrown light on various aspects of global terrorism, such as impacted countries, the types of attacks, the groups involved, etc.

Based on the terrorist attacks recorded world wide, Bomb explosions have been analyzed as the top attack modes. The project provides information on terrorism related attacks in Major Indian cities. It also offers valuable statistics on most impacted years and count of lost lives due to global terrorism.

It can be finally derived that terrorist attacks have significantly increased in the last 5 years and continues to be a threat to economy, worldwide.