**H1b Application –Statistical Analysis**

* Analysis on H1B is an employment-based, non-immigrant visa category for temporary foreign workers in the United States.
* We will be performing analysis on the H1B visa applicants between the years 2011-2015.
* It is the purpose of gathering information about the below mentioned fields using Apache Hadoop Tools.
* The dataset has nearly 3 million records.
* Purpose of this Analysis is to produce the Analytic overview of Applicants applying for visa process.

MAP-REDUCE

1.Is the number of petitions with Data Engineer job title increasing over time?

Program-

**import java.io.IOException;**

**import org.apache.hadoop.conf.Configuration;**

**import org.apache.hadoop.fs.Path;**

**import org.apache.hadoop.io.IntWritable;**

**import org.apache.hadoop.io.LongWritable;**

**import org.apache.hadoop.io.Text;**

**import org.apache.hadoop.mapreduce.Job;**

**import org.apache.hadoop.mapreduce.Mapper;**

**import org.apache.hadoop.mapreduce.Reducer;**

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

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

**public class data1 {**

**public static class datamap extends Mapper<LongWritable,Text,Text,IntWritable> {**

**public void map(LongWritable key,Text value,Context context)throws IOException,InterruptedException {**

**String[] word = value.toString().split("\t");**

**String word1 = word[4] +','+ word[8] +','+ word[7];**

**if(word[4].contains("DATA ENGINEER")) {**

**context.write(new Text(word1), new IntWritable(1));**

**}**

**}**

**}**

**public static class datared extends Reducer<Text,IntWritable,Text,IntWritable> {**

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

**int count=0;**

**for (IntWritable i :values)**

**{**

**count++;**

**}**

**context.write(key,new IntWritable(count));**

**}**

**}**

**public static void main(String args[]) throws IOException, ClassNotFoundException, InterruptedException{**

**Configuration conf= new Configuration();**

**Job job=Job.getInstance(conf, "");**

**job.setJarByClass(data1.class);**

**job.setMapperClass(datamap.class);**

**job.setReducerClass(datared.class);**

**job.setNumReduceTasks(1);**

**job.setMapOutputKeyClass(Text.class);**

**job.setMapOutputValueClass(IntWritable.class);**

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

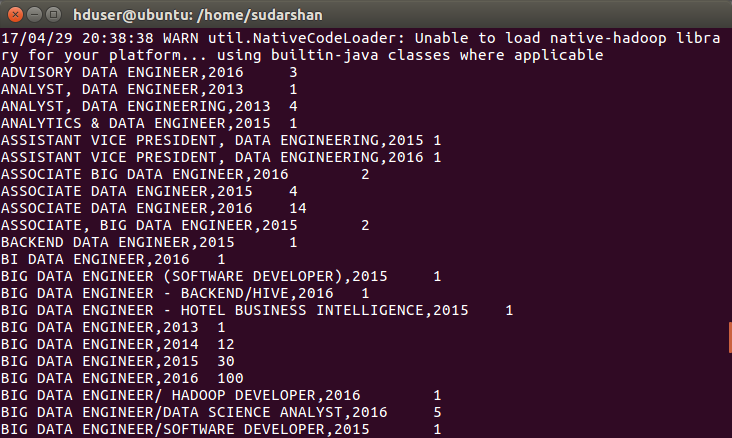
**FileOutputFormat.setOutputPath(job,new Path(args[1]));**

**System.exit(job.waitForCompletion(true) ? 0 :1);**

**}**

**}**

OUTPUT-



2.Which part of the US has the most Data Engineer jobs for each year?

Program-

**import java.io.IOException;**

**import java.util.TreeMap;**

**import org.apache.hadoop.conf.Configuration;**

**import org.apache.hadoop.fs.Path;**

**import org.apache.hadoop.io.IntWritable;**

**import org.apache.hadoop.io.LongWritable;**

**import org.apache.hadoop.io.NullWritable;**

**import org.apache.hadoop.io.Text;**

**import org.apache.hadoop.mapreduce.Job;**

**import org.apache.hadoop.mapreduce.Mapper;**

**import org.apache.hadoop.mapreduce.Reducer;**

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

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

**public class data2 {**

**public static class datamap extends Mapper<LongWritable,Text,Text,IntWritable> {**

**public void map(LongWritable key,Text value,Context context)throws IOException,InterruptedException {**

**String[] word = value.toString().split("\t");**

**String word1 = word[4] +','+ word[8] +','+ word[7];**

**if(word[4].contains("DATA ENGINEER"))**

**{**

**context.write(new Text(word1), new IntWritable(1));**

**}**

**}**

**}**

**public static class datared extends Reducer<Text,IntWritable,NullWritable,Text>**

**{**

**private TreeMap<Integer,Text> repToRecordMap= new TreeMap<Integer,Text>();**

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

**{**

**int count=0;**

**String myvalue= "";**

**for(IntWritable val:values)**

**{**

**count++;**

**}**

**myvalue= key.toString();**

**String mycount= String.format("%d", count);**

**myvalue= myvalue + ',' + mycount;**

**repToRecordMap.put(new Integer(mycount), new Text(myvalue));**

**if(repToRecordMap.size() >10)**

**{**

**repToRecordMap.remove(repToRecordMap.firstKey());**

**}**

**}**

**protected void cleanup(Context context) throws IOException, InterruptedException**

**{**

**for(Text t: repToRecordMap.values())**

**{**

**context.write(NullWritable.get(), t);**

**}**

**}**

**}**

**public static void main(String args[]) throws IOException, ClassNotFoundException, InterruptedException{**

**Configuration conf = new Configuration();**

**Job job = Job.getInstance(conf, "data count");**

**job.setJarByClass(data2.class);**

**job.setMapperClass(datamap.class);**

**job.setReducerClass(datared.class);**

**job.setNumReduceTasks(1);**

**job.setMapOutputKeyClass(Text.class);**

**job.setMapOutputValueClass(IntWritable.class);**

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

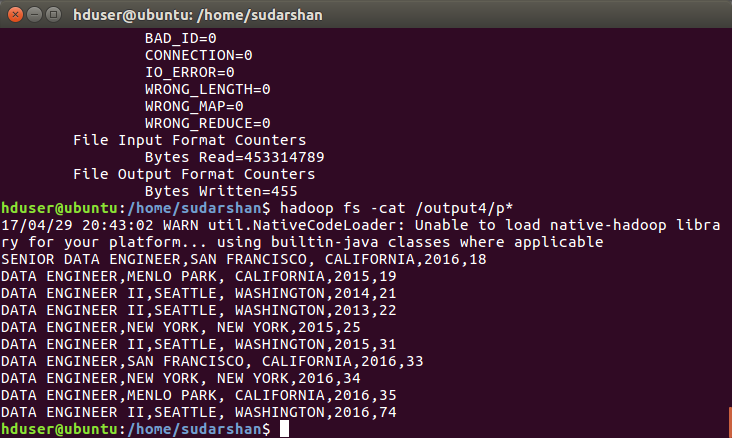
**FileOutputFormat.setOutputPath(job, new Path(args[1]));**

**System.exit(job.waitForCompletion(true) ? 0 :1);**

**}**

**}**

**OUTPUT-**

****

3.Which industry has the most number of Data Scientist positions.

Program-

**import java.io.IOException;**

**import java.util.TreeMap;**

**import org.apache.hadoop.conf.Configuration;**

**import org.apache.hadoop.fs.Path;**

**import org.apache.hadoop.io.IntWritable;**

**import org.apache.hadoop.io.LongWritable;**

**import org.apache.hadoop.io.NullWritable;**

**import org.apache.hadoop.io.Text;**

**import org.apache.hadoop.mapreduce.Job;**

**import org.apache.hadoop.mapreduce.Mapper;**

**import org.apache.hadoop.mapreduce.Reducer;**

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

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

**public class data3 {**

**public static class datamap extends Mapper<LongWritable,Text,Text,IntWritable> {**

**public void map(LongWritable key,Text value,Context context)throws IOException,InterruptedException {**

**String[] word = value.toString().split("\t");**

**String word1 = word[4] +','+ word[2];**

**if(word[4].contains("DATA SCIENTIST"))**

**{**

**context.write(new Text(word1), new IntWritable(1));**

**}**

**}**

**}**

**public static class datared extends Reducer<Text,IntWritable,NullWritable,Text>**

**{**

**private TreeMap<Integer,Text> repToRecordMap= new TreeMap<Integer,Text>();**

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

**{**

**int count=0;**

**String myvalue= "";**

**for(IntWritable val:values)**

**{**

**count++;**

**}**

**myvalue= key.toString();**

**String mycount= String.format("%d", count);**

**myvalue= myvalue + ',' + mycount;**

**repToRecordMap.put(new Integer(mycount), new Text(myvalue));**

**if(repToRecordMap.size() >1)**

**{**

**repToRecordMap.remove(repToRecordMap.firstKey());**

**}**

**}**

**protected void cleanup(Context context) throws IOException, InterruptedException**

**{**

**for(Text t: repToRecordMap.values())**

**{**

**context.write(NullWritable.get(), t);**

**}**

**}**

**}**

**public static void main(String args[]) throws IOException, ClassNotFoundException, InterruptedException{**

**Configuration conf = new Configuration();**

**Job job = Job.getInstance(conf, "data count");**

**job.setJarByClass(data3.class);**

**job.setMapperClass(datamap.class);**

**job.setReducerClass(datared.class);**

**job.setNumReduceTasks(1);**

**job.setMapOutputKeyClass(Text.class);**

**job.setMapOutputValueClass(IntWritable.class);**

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

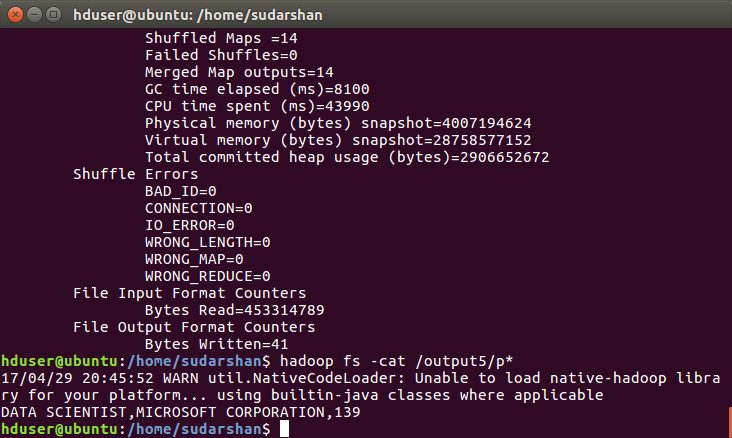
**FileOutputFormat.setOutputPath(job, new Path(args[1]));**

**System.exit(job.waitForCompletion(true) ? 0 :1);**

**}**

**}**

**OUTPUT-**

****

4.Find the most popular top 10 job positions for H1B visa applications for each year?

Program-

**import java.io.IOException;**

**import java.util.TreeMap;**

**import org.apache.hadoop.conf.Configuration;**

**import org.apache.hadoop.fs.Path;**

**import org.apache.hadoop.io.IntWritable;**

**import org.apache.hadoop.io.LongWritable;**

**import org.apache.hadoop.io.NullWritable;**

**import org.apache.hadoop.io.Text;**

**import org.apache.hadoop.mapreduce.Job;**

**import org.apache.hadoop.mapreduce.Mapper;**

**import org.apache.hadoop.mapreduce.Reducer;**

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

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

**public class top10positions**

**{**

**public static class Top10Mapper extends**

**Mapper<LongWritable, Text, Text, IntWritable> {**

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

**) throws IOException, InterruptedException {**

**final IntWritable one= new IntWritable(1);**

**try {**

**String[] str = value.toString().split("\t");**

**String position= str[4];**

**int year= Integer.parseInt(str[7]);**

**String status= str[1];**

**String myyear= String.format("%d", year);**

**String myvalue= position + ',' + status + ',' + myyear;**

**context.write(new Text(myvalue),one);**

**}**

**catch(Exception e)**

**{**

**System.out.println(e.getMessage());**

**}**

**}**

**}**

**public static class Top10Reducer extends**

**Reducer<Text, IntWritable, NullWritable, Text> {**

**private TreeMap<Integer, Text> repToRecordMap = new TreeMap<Integer, Text>();**

**public void reduce(Text key, Iterable<IntWritable> values,**

**Context context) throws IOException, InterruptedException {**

**long count=0;**

**String myvalue= "";**

**String mycount= "";**

**for (IntWritable val : values) {**

**//String[] token= key.toString().split(",");**

**count++;**

**}**

**myvalue= key.toString();**

**mycount= String.format("%d", count);**

**myvalue= myvalue + ',' + mycount;**

**repToRecordMap.put(new Integer(mycount), new Text(myvalue));**

**if (repToRecordMap.size() > 10) {**

**repToRecordMap.remove(repToRecordMap.firstKey());**

**}**

**}**

**protected void cleanup(Context context) throws IOException,**

**InterruptedException {**

**for (Text t : repToRecordMap.descendingMap().values()) {**

**// Output our five records to the file system with a null key**

**context.write(NullWritable.get(), t);**

**}**

**}**

**}**

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

**Configuration conf = new Configuration();**

**Job job = Job.getInstance(conf, "Top 10 positions");**

**job.setJarByClass(top10positions.class);**

**job.setMapperClass(Top10Mapper.class);**

**job.setReducerClass(Top10Reducer.class);**

**job.setMapOutputKeyClass(Text.class);**

**job.setMapOutputValueClass(IntWritable.class);**

**job.setOutputKeyClass(NullWritable.class);**

**job.setOutputValueClass(Text.class);**

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

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

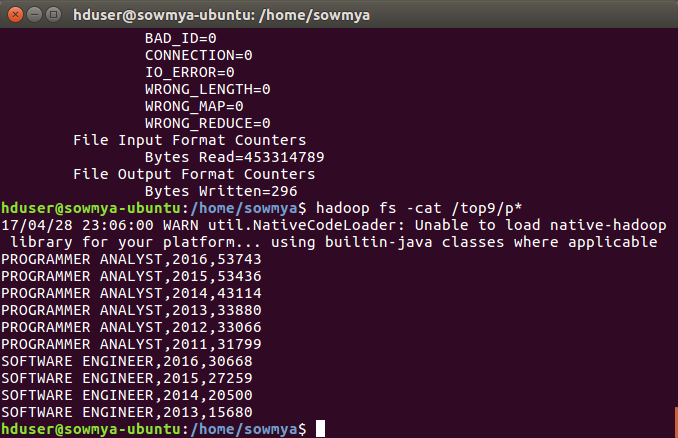
**System.exit(job.waitForCompletion(true) ? 0 : 1);**

**}**

**}**

**}**

**Output-**

****

**HIVE**

**LOADING IN HIVE-**

CREATE TABLE h1b\_applications(s\_no int,case\_status string, employer\_name string, soc\_name string, job\_title string, full\_time\_position string,prevailing\_wage int,year string, worksite string,worksite2 string,longitute double, latitute double )

ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.OpenCSVSerde'

WITH SERDEPROPERTIES (

"separatorChar" = ",",

"quoteChar" = "\""

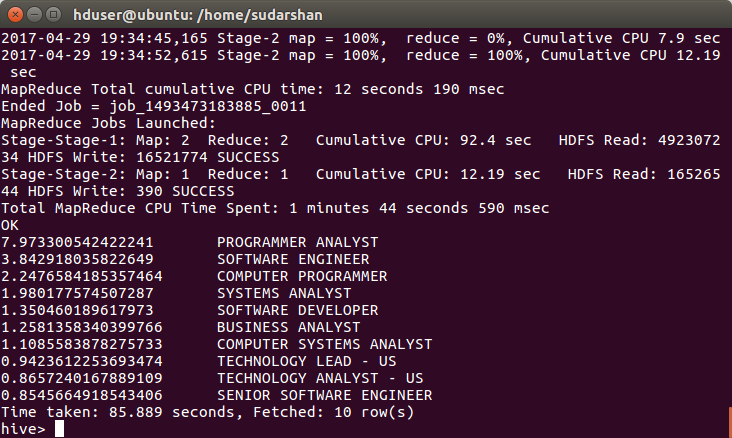
) STORED AS TEXTFILE;

1.Which are the top 10 job positions which have the highest success rate in petitions?

Query-

**select (count(s\_no)/t.tot)\*100 as ctn,job\_title from pro,(select count(s\_no) as tot from pro)t where case\_status ='CERTIFIED' or case\_status= 'CERTIFIED-WITHDRAWN' group by job\_title order by ctn desc limit 10;**

output-

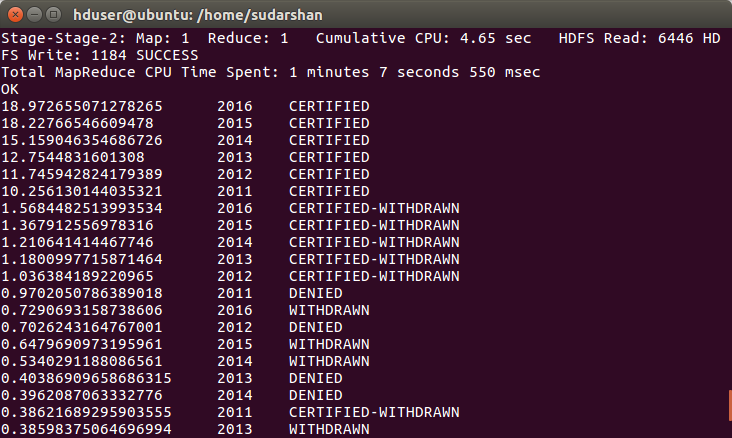


2. Find the percentage and the count of each case status on total applications for each year. Create a graph depicting the pattern of All the cases over the period of time

Query-

**select (count(s\_no)/t.tot)\*100 as ctn,year,case\_status from pro,(select count(s\_no) as tot from pro)t group by case\_status,year order by ctn desc;**

OUTPUT-

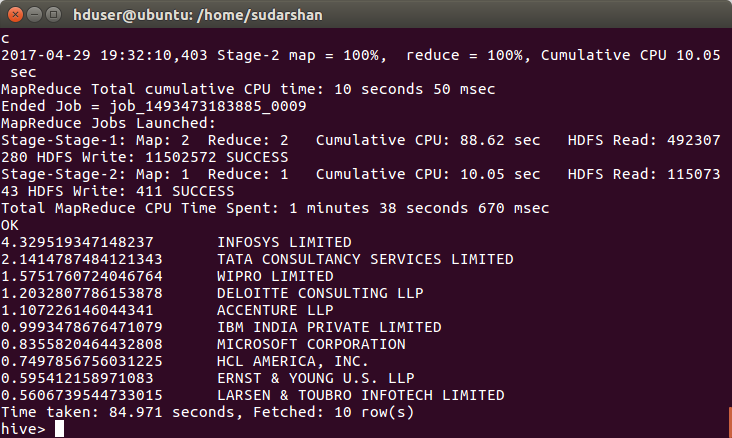


3. Which are top ten employers who have the highest success rate in petitions

Query –

**select (count(s\_no)/t.tot)\*100 as ctn, employer\_name from pro,(select count(s\_no) as tot from pro)t where case\_status ='CERTIFIED' or case\_status= 'CERTIFIED-WITHDRAWN' group by employer\_name order by ctn desc limit 10;**

output-



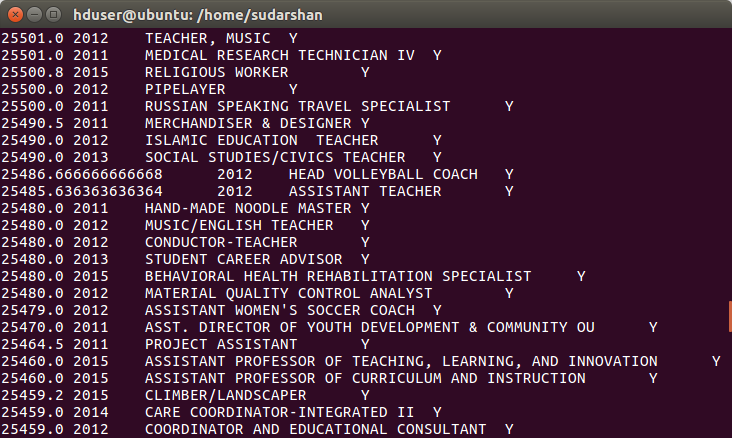
4. Find the average Prevailing Wage for each Job for each Year (take part time and full time separate)

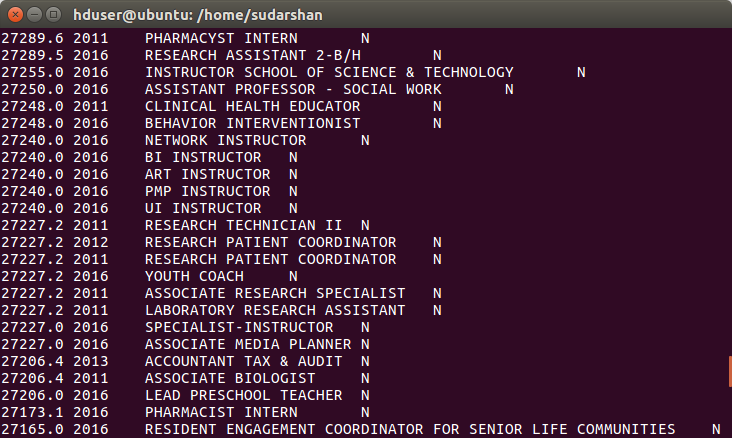
Query-

select avg(prevailing\_wage) as ctn,year,job\_title,full\_time\_position from pro where full\_time\_position='Y' group by year,job\_title,full\_time\_position order by ctn desc;

select avg(prevailing\_wage) as ctn,year,job\_title,full\_time\_position from pro where full\_time\_position='N' group by year,job\_title,full\_time\_position order by ctn desc;

output-





PIG TOOL-

Loading DATA in Grunt shell-

a= load '/home/sudarshan/Downloads/project1/mapreduce' using PigStorage('\t') AS (s\_no:int,case\_status:chararray,employer\_name:chararray,soc\_name:chararray,job\_title:chararray,full\_time\_position:chararray,prevailing\_wage,year,worksite1:chararray,longitute,lattitute);

1.Find top 5 job titles who are having highest growth in applications.

Script-

gp= group a by (year,job\_title);

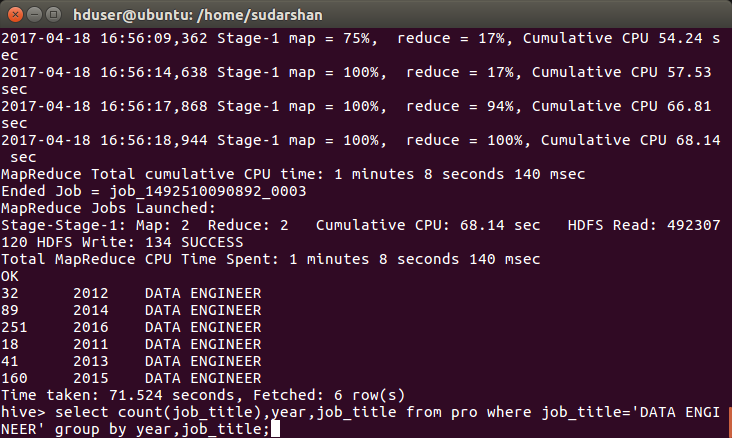
ct= foreach gp generate group as year, COUNT(a) as headcount;

obc= order ct by $1 desc;

limit2= limit obc 5;

Dump limit2;

output-



2. Which top 5 employers file the most petitions each year?

Script-

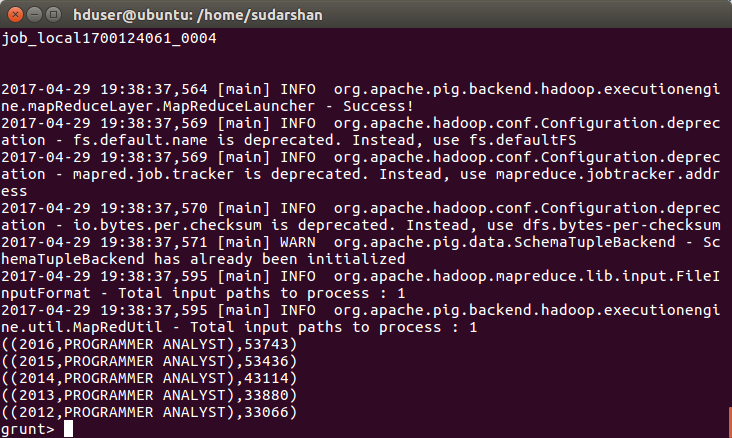
gp= group a by (year,employer\_name);

ct= foreach gp generate group as year, COUNT(a) as headcount;

obc= order ct by $1 desc;

limit2= limit obc 5;

output-



3. Create a bar graph to depict the number of applications for each year

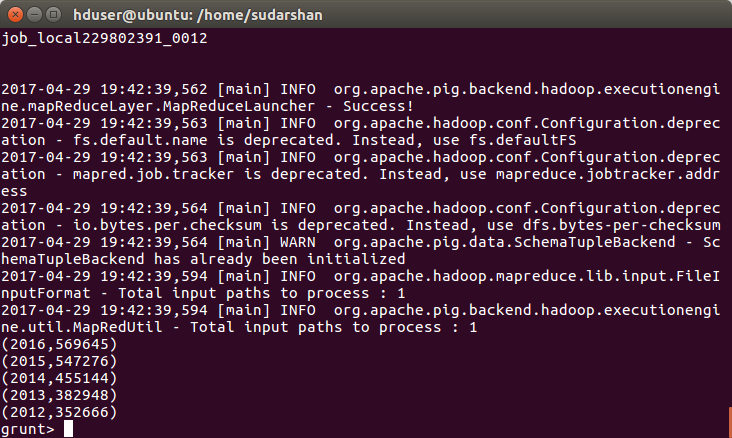
Script-

gp= group a by (year);

ct= foreach gp generate group as year, COUNT(a) as headcount;

obc= order ct by $1 desc;

limit2= limit obc 5;

output-  


4. find top 5 locations in the US who have got certified visa for each year.

Script-

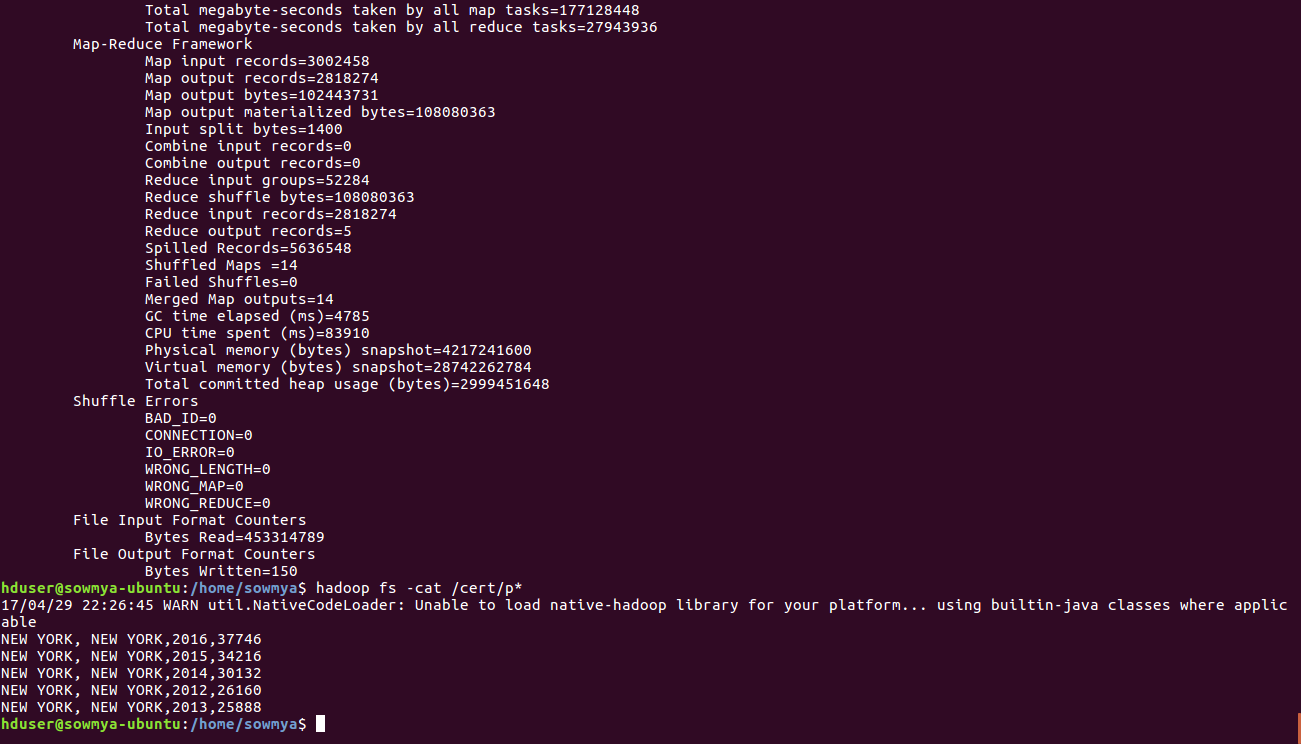
filter\_bag= filter a by case\_status=='CERTIFIED';

group\_all= group filter\_bag by (year);

count\_all= foreach group\_all generate $0, COUNT(filter\_bag) as headcount;

orderby= order count\_all by headcount desc;

limit3= limit orderby 5;

output-

**DATA EXPORT TO RELATIONAL DATABASE**

The OUTPUT data can be used whenever needed, apart from HDFS, it is better to export data to the local database where analysts do simple query in MySQL and do analysis in future.

This can be achieved using Apache Sqoop,

* Sqoop is used to import and export data from or to Relational Databases
* HDFS file can be directly exported to RDBMS



Make an input file for the top 10 job positions who have highest success rate in petitions and put it in HDFS. The command for it:

[hduser@sudarshan-ubuntu](mailto:hduser@sowmya-ubuntu):~$ hadoop fs -put /home/sudarshan/visa.txt /niit

**STEPS FOR EXPORT:**

Enter the mysql server: mysql -u root -p

* Create a database in Mysql.
* **Create a table in Mysql to export data from HDFS.** The table should be as exact schema as of HDFS file. Command is as follows:

create table h1b\_app(

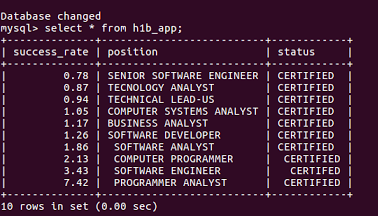
success\_rate FLOAT NOT NULL,

position VARCHAR(40) NOT NULL,

status VARCHAR(40) NOT NULL,

PRIMARY KEY(success\_rate));

Output-



ANALYSIS FROM THE REPORT:

Analysis of H1-B US-Visa is as follows:

* The number of applications submitted for H1-B Visas are increasing repidly every year. So the government needs to select the top candidates
* The top job positions who get certified are Programmer Analyst, Software Enginner and Computer Programmer
* The top companies who get certified are Infosys Ltd, TCS Ltd and Wipro Ltd.

SUGGESTIONS FROM THE REPORT:

* With Trump's decision on H-1B, Indian IT companies may suffer a major setback as top companies offering H-1B visa were majorly outsourcing employment to India. It will hit hard the entry level techies. So Young People should really work hard for H1-B Visa
* There is a setback for Computer Programmers according to the new Policies of H1-B VISA. The competition has increased in Engineering Field.