PROJECT REPORT

H1 B APPLICATIONS

Submitted by

KISHAN L R

S181113400068

ABSTRACT

The H-1B is a non-immigrant visa in the United States under the Immigration and Nationality Act, section 101(a) (17) (H). It allows U.S. employers to temporarily employ foreign workers in specialty occupations. If a foreign worker in H-1B status quits or is dismissed from the sponsoring employer, the worker must either apply for and be granted a change of status to another non-immigrant status, find another employer (subject to application for adjustment of status and/or change of visa), or leave the United States. Effective January 17, 2017, USCIS modified the rules to allow a grace period of up to 60 days. This topic is of international importance. Data analysis on this vast topic can give valuable information.

INTRODUCTION

The H1B is an employment-based, non-immigrant visa category for temporary foreign workers in the United States. For a foreign national to apply for H1B visa, an US employer must offer a job and petition for H1B visa with the US immigration department. This is the most common visa status applied for and held by international students once they complete college/ higher education (Masters, Ph.D.) and work in a full-time position. The U.S. Department of Labor (DOL) is responsible for ensuring that foreign workers do not displace or adversely affect wages or working conditions of U.S. workers. For every H-1B petition filed with the USCIS, there must be included a Labor Condition Application (LCA) (not to be confused with the labor certification), certified by the U.S. Department of Labor. The LCA is designed to ensure that the wage offered to the non-immigrant worker meets or exceeds the "prevailing wage" in the area of employment. The LCA also contains an attestation section designed to prevent the program from being used to import foreign workers to break a strike or replace U.S. citizen workers. While an employer is not required to advertise the position before hiring an H-1B non-immigrant pursuant to the H-1B visa approval, the employer must notify the employee representative about the Labor Condition Application (LCA)—or if there is no such representation, the employer must publish the LCA at the workplace and the employer's office. Under the regulations, LCAs are a matter of public record. Corporations hiring H-1B workers are required to make these records available to any member of the public who requests to look at them. Copies of the relevant records are also available from various web sites, including the Department of Labor.

OBJECTIVES

1. Data collection and production of information for government ministries and local authorities, for budgeting purposes.

2. Production of information which serves bodies, organizations and various other elements in the fields of education, the economy, business, research, etc.

3. Decision-making that facilitates the development of socio-economic policies -enhance the welfare of the population.

4. Processing and analyzing large amount of raw data by using map-reduce programming model and distributed computing on HADOOP framework to improve time and complexity.

BIG DATA

Big data is a term that describes the large volume of data – both structured and unstructured – that inundates a business on a day-to-day basis. But it's not the amount of data that's important. It's what organizations do with the data that matters. Big data can be analyzed for insights that lead to better decisions and strategic business moves.

While the term "big data" is relatively new, the act of gathering and storing large amounts of information for eventual analysis is ages old. The concept gained momentum in the early 2000s when industry analysts articulated the now-mainstream definition of big data as the five Vs:

Volume – Organizations collect data from a variety of sources, including business transactions, social media and information from sensor or machine-to-machine data. In the past, storing it would’ve been a problem – but new technologies (such as Hadoop) have eased the burden.

Velocity – Data streams in at an unprecedented speed and must be dealt with in a timely manner. RFID tags, sensors and smart metering are driving the need to deal with torrents of data in near-real time.

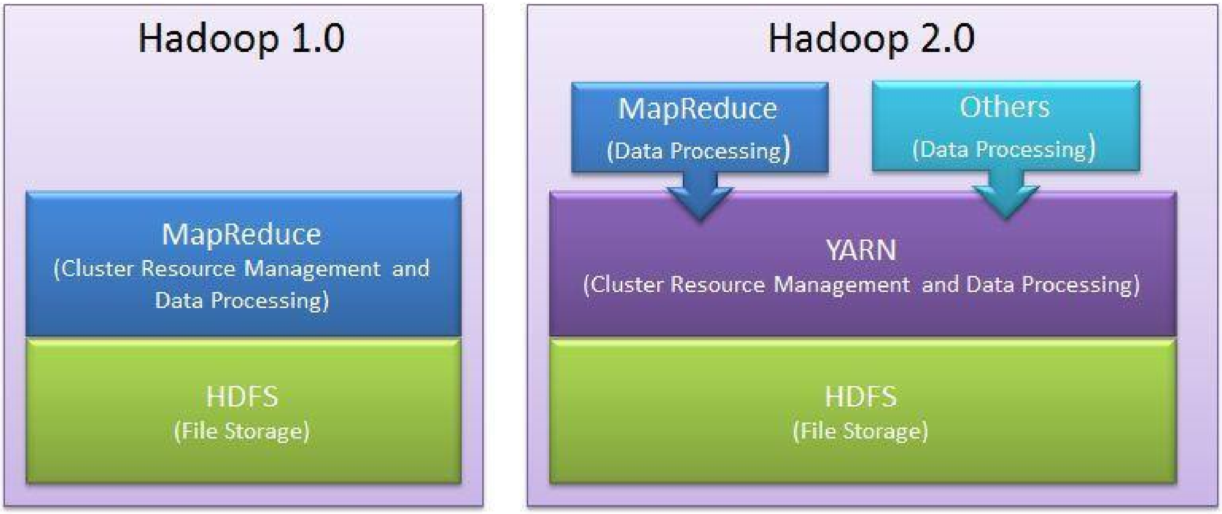
Variety – Data comes in all types of formats – from structured, numeric data in traditional databases to unstructured text documents, email, video, audio, stock ticker data and financial transactions.

Veracity – Refers to the biases, noise and abnormality in data. Is the data that is being stored, and mined meaningful to the problem being analyzed.

Value: Then there is another V to take into account when looking at Big Data: Value! It is all well and good having access to big data but unless we can turn it into value it is useless. So you can safely argue that 'value' is the most important V of Big Data. It is important that businesses make a business case for any attempt to collect and leverage big data. It is so easy to fall into the buzz trap and embark on big data initiatives without a clear understanding of costs and benefits.

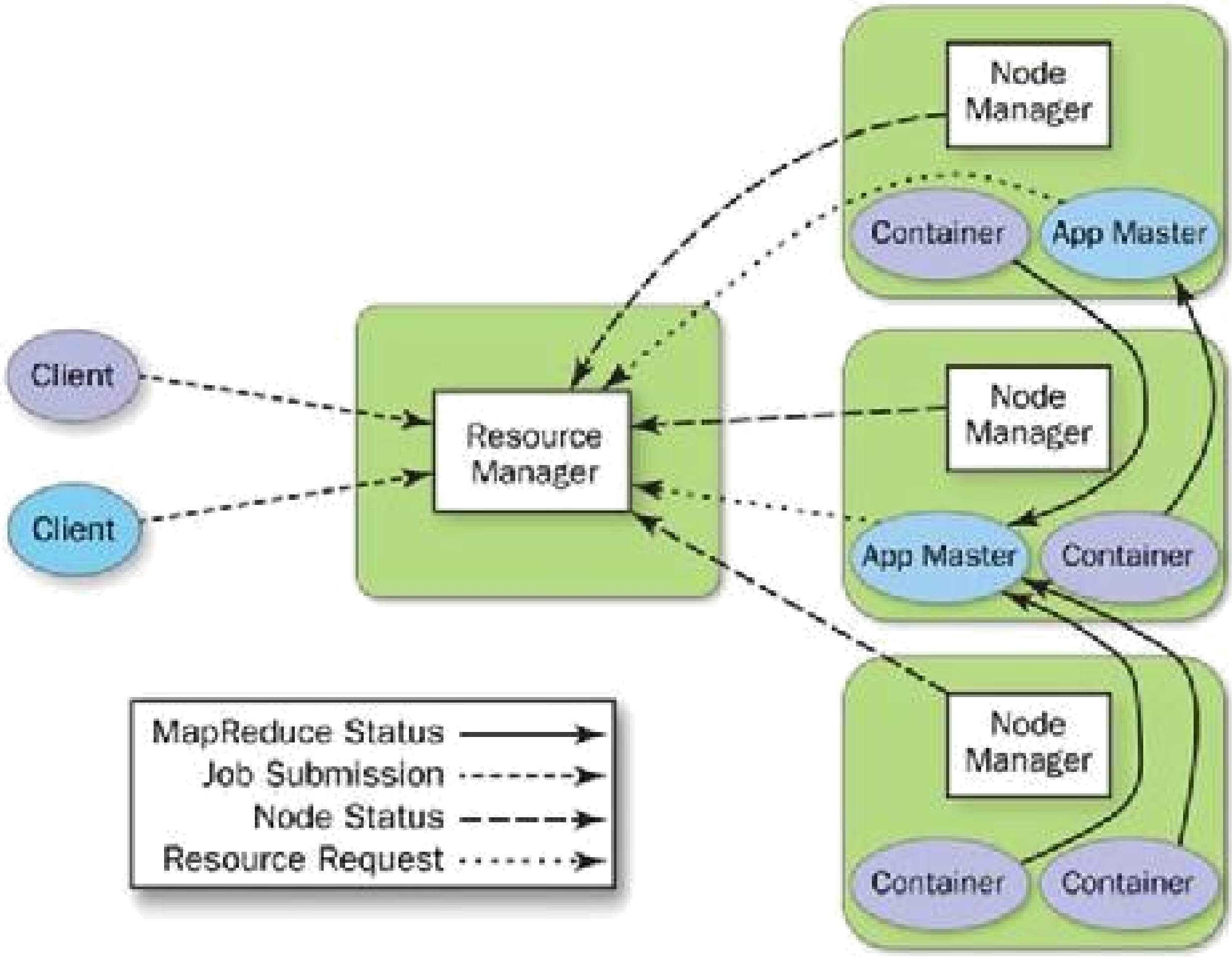
HADOOP ARCHITECTURE

Apache Hadoop is an open-source software framework for storage and large-scale processing of data-sets on clusters of commodity hardware. There are mainly five building blocks inside this runtime environment (from bottom to top):



The cluster is the set of host machines (nodes). Nodes may be partitioned in racks. This is the hardware part of the infrastructure.

The YARN Infrastructure (Yet another Resource Negotiator) is the framework responsible for providing the computational resources (e.g., CPUs, memory, etc.) needed for application executions. Important element is:



Resource Manager

TOOLS USED

MapReduce: MapReduce is a processing technique and a program model for distributed computing based on java. The MapReduce algorithm contains two important tasks, namely Map and Reduce. Map takes a set of data and converts it into another set of data, where individual elements are broken down into tuples (key/value pairs). Secondly, reduce task, which takes the output from a map as an input and combines those data tuples into a smaller set of tuples. As the sequence of the name MapReduce implies, the reduce task is always performed after the map job.

Apache Hive: Apache Hive is an open-source data warehouse system for querying and analyzing large datasets stored in Hadoop files. It was developed at Facebook initially for querying their huge datasets. Hadoop is a framework for handling large datasets in a distributed computing environment.

Apache Pig: Apache Pig is a platform for analyzing large data sets that consists of a high-level language for expressing data analysis programs, coupled with infrastructure for evaluating these programs. The salient property of Pig programs is that their structure is amenable to substantial parallelization, which in turns enables them to handle very large data sets.

H1B APPLICATIONS

Sample Data:

	CASE_STATUS	EMPLOYER_NAME	SOC_NAME	JOB_TITLE	FULL_TIME_POSITION	PREVAILING_WAGE	YEAR	WORKSITE	lon	lat
1	CERTIFIED-WITHDRAWN	UNIVERSITY OF MICHIGAN	BIOCHEMISTS AND BIOPHYSICISTS	POSTDOCTORAL RESEARCH FELLOW	N	36067	2016	ANN ARBOR, MICHIGAN	-83.743	42.28083
2	CERTIFIED-WITHDRAWN	GOODMAN NETWORKS, INC.	CHIEF EXECUTIVES	CHIEF OPERATING OFFICER	Y	242674	2016	PLANO, TEXAS	-96.6989	33.01984
3	CERTIFIED-WITHDRAWN	PORTS AMERICA GROUP, INC.	CHIEF EXECUTIVES	CHIEF PROCESS OFFICER	Y	193066	2016	JERSEY CITY, NEW JERSEY	-74.0776	40.72816
4	CERTIFIED-WITHDRAWN	GATES CORPORATION, A WHOLLY-OWNED SUBSIDIARY OF TOMKINS PLC	CHIEF EXECUTIVES	REGIONAL PRESIDEN, AMERICAS	Y	220314	2016	DENVER, COLORADO	-104.99	39.73924
5	WITHDRAWN	PEABODY INVESTMENTS CORP.	CHIEF EXECUTIVES	PRESIDENT MONGOLIA AND INDIA	Y	157518.4	2016	ST. LOUIS, MISSOURI	-90.1994	38.627

Data set column description.

1. **CASE_STATUS**: Status associated with the last significant event or decision. Valid values include.

   * *"Certified"*: Employer filed the LCA, which was approved by DOL.

   * *"Certified-Withdrawn"*: LCA was approved but later withdrawn by employer.

   * *"Denied"*: LCA was denied by DOL.

   * *"Withdrawn"*: LCA was withdrawn by employer before.

2. **EMPLOYER_NAME**: Name of employer submitting labor condition application.

3. **SOC_NAME**: the Occupational name associated with the SOC_CODE. SOC_CODE is the occupational code associated with the job being requested for temporary labor condition, as classified by the Standard Occupational Classification (SOC) System.

4. **JOB_TITLE**: Title of the job: FULL_TIME_POSITION

   * *Y* = Full Time Position.

   * *N* = Part Time Position.

5. **PREVAILING_WAGE**: Prevailing Wage for the job being requested for temporary labor condition. The wage is listed at annual scale in USD. The prevailing wage for a job position is defined as the average wage paid to similarly employed workers in the requested occupation in the area of intended employment. The prevailing wage is based on the employer's minimum requirements for the position.

6. **YEAR**: Year in which the H1B visa petition was filed.

7. **WORKSITE**: City and State information of the foreign worker's intended area of employment.

8. **lon**: Longitude of the Worksite.

9. **lat**: Latitude of the Worksite.

ANALYSIS

Analysis carried out are:

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

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

3) Which part of the US has the most Data Engineer jobs for each year?

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

5) Which industry has the most number of Data Scientist positions?

6) Which top 5 employers file the most petitions each year?

7) Find the most popular top 10 job positions for H1B visa applications for each year?

8) 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.

9) Create a bar graph to depict the number of applications for each year

10) find the average Prevailing Wage for each Job for each Year (take part time and full time separate).Arrange the output in descending order.

11) Which are employers along with the number of petitions who have the success rate more than 70% in petitions and total petitions filed more than 1000?

12) Which are the job positions along with the number of petitions which have the success rate more than 70% in petitions and total petitions filed more than 1000?

Source code and results of analysis can be found at Github.

A bash menu is developed to select and analyze the required problem

```
#!/bin/bash
show_menu()
{
    NORMAL=`echo "\033[m"`
    MENU=`echo "\033[36m"` #Blue
    NUMBER=`echo "\033[33m"` #yellow
    FGRED=`echo "\033[41m"`
    RED_TEXT=`echo "\033[31m"`
    ENTER_LINE=`echo "\033[33m"`
    echo -e "${MENU}**********************H1B
APPLICATIONS***********************${NORMAL}"
    echo -e "${MENU}${NUMBER} 1) ${MENU} Is the number of petitions with
Data Engineer job title increasing over time?${NORMAL}"
    echo -e "${MENU}${NUMBER} 2) ${MENU} Find top 5 job titles who are
having highest growth in applications. ${NORMAL}"
    echo -e "${MENU}${NUMBER} 3) ${MENU} Which part of the US has the most
Data Engineer jobs for each year? ${NORMAL}"
    echo -e "${MENU}${NUMBER} 4) ${MENU} find top 5 locations in the US who
have got certified visa for each year.${NORMAL}"
    echo -e "${MENU}${NUMBER} 5) ${MENU} Which industry has the most number
of Data Scientist positions?${NORMAL}"
    echo -e "${MENU}${NUMBER} 6) ${MENU} Which top 5 employers file the most
petitions each year? ${NORMAL}"
    echo -e "${MENU}${NUMBER} 7) ${MENU} Find the most popular top 10 job
positions for H1B visa applications for each year?${NORMAL}"
    echo -e "${MENU}${NUMBER} 8) ${MENU} Find the percentage and the count
of each case status on total applications for each year. Create a graph
depicting the pattern of All the cases over the period of time.${NORMAL}"
    echo -e "${MENU}${NUMBER} 9) ${MENU} Create a bar graph to depict the
number of applications for each year${NORMAL}"
    echo -e "${MENU}${NUMBER} 10) ${MENU}Find the average Prevailing Wage
for each Job for each Year (take part time and full time separate) arrange
output in descending order${NORMAL}"
    echo -e "${MENU}${NUMBER} 11) ${MENU} Which are employers who have the
highest success rate in petitions more than 70% in petitions and total
petions filed more than 1000?${NORMAL}"
    echo -e "${MENU}${NUMBER} 12) ${MENU} Which are the top 10 job positions
which have the  success rate more than 70% in petitions and total petitions
filed more than 1000? ${NORMAL}"
    echo -e "${MENU}${NUMBER} 13) ${MENU}Export result for option no 12 to
MySQL database.${NORMAL}"
    echo -e "${MENU}*********************************************${NORMAL}"
    echo -e "${ENTER_LINE}Please enter a menu option and enter or $
{RED_TEXT}enter to exit. ${NORMAL}"
    read opt
}
function option_picked()
{
    COLOR='\033[01;31m' # bold red
    RESET='\033[00;00m' # normal white
    MESSAGE="$1"  #modified to post the correct option selected
    echo -e "${COLOR}${MESSAGE}${RESET}"
}
clear
start-all.sh | zenity --progress --width 150 --title="Hadoop Services
```

```
Starting" --pulsate --auto-close #--percentage
yad --info --title="Project" --text '<span foreground="red"
font="14">\t\t\tWelcome To BigData Project\n</span><span
font="12">\n<b>\tAnalysis And Summarization Of H1B Applicants</b>\n</span>'
--width=450 --height=10 --button="gtk-cancel:252" --button="gtk-ok:0"
--center --timeout 3
show_menu


while [ opt != '' ]
do
      if [[ $opt = "" ]]; then
            exit;
      else
      #start-dfs.sh
      #start-yarn.sh
      #Pig/start-jobhistory.sh
      #sleep 6

        case $opt in
            1) clear;
                  option_picked "1) Is the number of petitions with Data
Engineer job title increasing over time?";
                  hadoop fs -rmr /niit/projout1
                  hadoop fs -rmr /niit/projout1
                  rm -r /home/hduser/h1bproject/projectout/1a
                  hadoop jar /home/hduser/h1bproject/proj.jar
project/DataEngineerJob /niit/h1b/0* /niit/projout1 | zenity --progress
--pulsate --title="Job Running" --auto-close
                  echo -e "\n1a) Is the number of petitions with Data
Engineer job title increasing over time?\n\n"
                  hadoop fs -get /niit/projout1
/home/hduser/h1bproject/projectout/1a
                  hadoop fs -cat /niit/projout1/p*
                  sleep 5
                  show_menu;
            ;;
            2) clear;
                  option_picked "2) Find top 5 job titles who are having
highest growth in applications. ";
                  rm -r /home/hduser/h1bproject/projectout/1b
                  pig -x local /home/hduser/h1bproject/pig/h1b1b.pig |
zenity --progress --title="Pig Job Running" --pulsate --auto-close
                  echo -e "\n1b) Find top 5 job titles who are having
highest growth in applications.?\n\n "
                  cat /home/hduser/h1bproject/projectout/1b/p*
                  sleep 5
                  show_menu;
            ;;
            3) clear;
                  while : ; do
                  option_picked "3) Which part of the US has the most Data
Engineer jobs for each year?\n";
                  #echo -e "Do you wish to see \n1. The entire result \n2.
Year wise result\n"
                  #echo -e "choose option 1 or 2 \n"
                  #read choice
                  choice=$(yad --title "Result Selection" --entry --text
'<span foreground="red" font="14">Do you wish to see 1. The entire result 2.
Year wise result\n</span><span font="12">\n<b>choose option 1 or 2
</b>\n</span>' --width=450 --height=100 --center --button="gtk-cancel:252"
--button="gtk-ok:0")
            #     while ([ $choice != '1'] || [ $choice != '2']) ;
            #
            #           echo -e "Do you wish to see 1. The entire result
\n2. Year wise result\n"
            #           read 'choose option 1 or 2 ' choice
                        case $choice in
                        1) clear;
                              hadoop fs -rmr /niit/out2a2
                              rm -r /home/hduser/h1bproject/projectout/2a
                              hadoop jar /home/hduser/h1bproject/proj.jar
project/WorksiteUSDataEngineerJob /niit/h1b/0* /niit/out2a2 ALL | zenity
--progress --title="Pig Job Running" --pulsate --auto-close
                              hadoop fs -get /niit/out2a2
/home/hduser/h1bproject/projectout/2a
                              echo -e "\n2a) Which part of the US has the
most Data Engineer jobs for every year?\n"
                              hadoop fs -cat /niit/out2a2/p*
                              sleep 5
                              break
                        ;;
                        2) clear;
                              echo -e "Enter the year
(2011,2012,2013,2014,2015,2016)"
                              rm -r /home/hduser/h1bproject/projectout/2a2
                              #read year
                              year=$(yad --title "Year Selection" --entry
--text '<span foreground="red" font="14">Enter the year
(2011,2012,2013,2014,2015,2016)\n</span><span font="12">\n<b>choose any each
year</b>\n</span>' --width=450 --height=100 --center --button="gtk-
cancel:252" --button="gtk-ok:0")
                              hive -f /home/hduser/h1bproject/hive/h1b2a.sql
-hiveconf year=$year | zenity --progress --title="Hive Job Running"
--pulsate --auto-close
                              echo -e "\n2a) Which part of the US has the
most Data Engineer jobs for each year?\n"
                              cat /home/hduser/h1bproject/projectout/2a2/0*
                              sleep 5
                              break

                        ;;
                        *) clear;
                        echo -e "Error Command...\n"
                        sleep 2
                        ;;
                        esac
                        done
                  show_menu;
            ;;
            4) clear;
                  option_picked "4) find top 5 locations in the US who have
got certified visa for each year.";
                  rm -r /home/hduser/h1bproject/projectout/2b
                  pig -x local /home/hduser/h1bproject/pig/h1b2b.pig |
zenity --progress --title="Pig Job Running" --pulsate --auto-close
                  echo -e "\n2b) find top 5 locations in the US who have got
certified visa for each year.\n"
                  cat /home/hduser/h1bproject/projectout/2b/p*
                  sleep 5
                  show_menu;
            ;;
            5) clear;
                  option_picked "5) Which industry has the most number of
Data Scientist positions?";
                  rm -r /home/hduser/h1bproject/projectout/3
                  hive -f /home/hduser/h1bproject/hive/h1b3.sql | zenity
--progress --title="Hive Job Running" --pulsate --auto-close
```

```
                  echo -e "\n3) Which industry has the most number of Data
Scientist positions?\n"
                  cat /home/hduser/h1bproject/projectout/3/0*
                  sleep 5
                  show_menu;
            ;;
            6) clear;
                  option_picked "6)Which top 5 employers file the most
petitions each year?";
                  rm -r /home/hduser/h1bproject/projectout/4
                  pig -x local /home/hduser/h1bproject/pig/h1b4.pig | zenity
--progress --title="Pig Job Running" --pulsate --auto-close
                  echo -e "\n4)Which top 5 employers file the most petitions
each year?\n"
                  cat /home/hduser/h1bproject/projectout/4/p*
                  sleep 5
                  show_menu;
            ;;
            7) clear;
                  option_picked "7) Find the most popular top 10 job
positions for H1B visa applications for each year?";
                  #echo -e "For All Applications Select 1 or For Certified
Applications Select 2"
                  #read sel
                  sel=$(yad --title "Application Selection" --entry --text
'<span foreground="red" font="14">For All Applications Select 1 or For
Certified Applications Select 2\n</span><span font="12">\n<b>choose option 1
or 2 </b>\n</span>' --width=450 --height=100 --center --button="gtk-
cancel:252" --button="gtk-ok:0")
                  if [ $sel == '1' ];
                  then
                        #echo -e "Do you wish to see 1. The entire result
\n2. Year wise result\n"
                        #read choice
                        choice=$(yad --title "Result Selection" --entry
--text '<span foreground="red" font="14">Do you wish to see 1. The entire
result 2. Year wise result\n</span><span font="12">\n<b>choose option 1 or 2
</b>\n</span>' --width=450 --height=100 --center --button="gtk-cancel:252"
--button="gtk-ok:0")
                        #while [$choice != '1'] || [$choice != '2']
                        #do
                        #     echo -e "Do you wish to see 1. The entire
result \n2. Year wise result\n"
                        #     read 'choose option 1 or 2 ' choice
                        #done
                        if [ $choice == '1' ];
                        then
                              hadoop fs -rmr /niit/projout5a
                              rm -r /home/hduser/h1bproject/projectout/5a
                              hadoop jar /home/hduser/h1bproject/proj.jar
project/Top10JobPositions /niit/h1b/0* /niit/projout5a | zenity --progress
--title="Job Running" --pulsate --auto-close
                              hadoop fs -get /niit/projout5a
/home/hduser/h1bproject/projectout/5a
                              echo -e "\n5a) Find the most popular top 10
job positions for H1B visa applications for every year?\n"
                              hadoop fs -cat /niit/projout5a/p*
                              sleep 5
                        else
                              #echo -e "Enter the year
(2011,2012,2013,2014,2015,2016)"
                              #read year
                              year=$(yad --title "Year Selection" --entry
--text '<span foreground="red" font="14">Enter the year
```

```
(2011,2012,2013,2014,2015,2016)\n</span><span font="12">\n<b>choose any each
year</b>\n</span>' --width=450 --height=100 --center --button="gtk-
cancel:252" --button="gtk-ok:0")
                              rm -r /home/hduser/h1bproject/projectout/5a1
                              hive -e "insert overwrite local directory
'/home/hduser/h1bproject/projectout/5a1' row format delimited FIELDS
TERMINATED BY '\t' select job_title,year,count(case_status) as temp from
h1b.h1b_final where year= '$year' group by job_title,year order by temp desc
limit 10;" | zenity --progress --title="Hive Job Running" --pulsate --auto-
close
                              #hive -f
/home/hduser/h1bproject/hive/h1b5a.sql -hiveconf year=$year
                              echo -e "\n5a) Find the most popular top 10
job positions for H1B visa applications for each year?\n";
                              cat /home/hduser/h1bproject/projectout/5a1/0*
                              sleep 5
                        fi
                  else
                        #echo -e "Do you wish to see 1. The entire result
\n2. Year wise result\n"
                        #echo -e "choose option 1 or 2"
                        #read choice
                        choice=$(yad --title "Result Selection" --entry
--text '<span foreground="red" font="14">Do you wish to see 1. The entire
result 2. Year wise result\n</span><span font="12">\n<b>choose option 1 or 2
</b>\n</span>' --width=450 --height=100 --center --button="gtk-cancel:252"
--button="gtk-ok:0")
                        if [ $choice == '1' ];
                        then
                              hadoop fs -rmr /niit/projout5b
                              rm -r /home/hduser/h1bproject/projectout/5b
                              hadoop jar /home/hduser/h1bproject/proj.jar
project/Top10CertifiedJobPositions /niit/h1b/0* /niit/projout5b | zenity
--progress --title="Job Running" --pulsate --auto-close
                              hadoop fs -get /niit/projout5b
/home/hduser/h1bproject/projectout/5b
                              echo -e "\n5b) Find the most popular top 10
certified job positions for H1B visa applications for every year?\n";
                              hadoop fs -cat /niit/projout5b/p*
                              sleep 5
                        else
                              #echo -e "Enter the year
(2011,2012,2013,2014,2015,2016)"
                              #read year
                              rm -r /home/hduser/h1bproject/projectout/5b1
                              year=$(yad --title "Year Selection" --entry
--text '<span foreground="red" font="14">Enter the year
(2011,2012,2013,2014,2015,2016)\n</span><span font="12">\n<b>choose any each
year</b>\n</span>' --width=450 --height=100 --center --button="gtk-
cancel:252" --button="gtk-ok:0")
                              #hive -e "select
job_title,case_status,year,count(case_status ) as temp from h1b_final where
year= '$year' and case_status like 'CERTIFIED' group by
job_title,case_status,year order by temp desc limit 10; "
                              hive -f /home/hduser/h1bproject/hive/h1b5b.sql
-hiveconf year=$year | zenity --progress --title="Hive Job Running"
--pulsate --auto-close
                              echo -e "\n5b) Find the most popular top 10
certified job positions for H1B visa applications for each year?\n";
                              cat /home/hduser/h1bproject/projectout/5b1/0*
                              sleep 5
                        fi
                  fi
            show_menu;
            ;;
            8) clear;
                  option_picked "8) Find the percentage and the count of
each case status on total applications for each year.";
                  rm -r /home/hduser/h1bproject/projectout/6
                  rm -r /home/hduser/h1bproject/graph/6/data/
                  rm /home/hduser/h1bproject/graph/6/h1b6graph.jpeg

                  pig -x local /home/hduser/h1bproject/pig/h1b6.pig | zenity
--progress --title="Pig Job Running" --pulsate --auto-close
                  echo -e "\n6) Find the percentage and the count of each
case status on total applications for each year.\n"
                  cat /home/hduser/h1bproject/projectout/6/p*
                  sleep 3
                  #gnuplot -e "set grid;set title 'Case Status on total
applications for each year ';set yrange [0:100];set xrange[2011:2017];set
xlabel 'Year';set ylabel 'Percentage';plot
'/home/hduser/graph/6/data/filtcer/part-r-00000' u 1:5 w lp t 'CERTIFIED' lt
rgb "#8B0000" lw 3 pt 6,"/home/hduser/graph/6/data/filtcerwith/part-r-00000"
u 1:5 w lp t 'CERTIFIED-WITHDRAWN' lt rgb "#00008B" lw 3 pt
6,"/home/hduser/graph/6/data/filtden/part-r-00000" u 1:5 w lp t 'DENIED' lt
rgb "#808000" lw 3 pt 6,"/home/hduser/graph/6/data/filtwith/part-r-00000" u
1:5 w lp t 'WITHDRAWN' lt rgb "#00FF00" lw 3 pt 6;pause 5;set terminal
jpeg;set output '/home/hduser/graph/6/h1b6graph.jpeg';replot;exit gnuplot"
                  #try to use gnuplot in file without using gnuplot -e
                  gnuplot /home/hduser/h1bproject/graph/6/gnu1.gp
                  sleep 2
                  show_menu;
            ;;
            9) clear;
                  option_picked "9) The number of applications for each
year";
                  rm /home/hduser/h1bproject/graph/7/h1b7.dat
                  rm /home/hduser/h1bproject/graph/7/h1b7graph.jpeg
                  hive -f /home/hduser/h1bproject/hive/h1b7.sql >>
/home/hduser/h1bproject/graph/7/h1b7.dat | zenity --progress --title="Hive
Job Running" --pulsate --auto-close
                  echo -e "\n7) The number of applications for each year\n"
                  cat /home/hduser/h1bproject/graph/7/h1b7.dat
                  sleep 3
                  #gnuplot -e "set style line 1 lc rgb 'grey30' ps 0 lt 1 lw
2;set style line 2 lc rgb 'grey70' lt 1 lw 2;set style fill solid 1.0 border
rgb 'grey30';plot '/home/hduser/graph/h1b7.dat' every ::1 u 0:2:
(0.7):xtic(1) w boxes;pause 5;set term png;set terminal png size 400,300;set
output '/home/hduser/graph/h1b7graph.png';replot;exit gnuplot"
                  gnuplot /home/hduser/h1bproject/graph/7/gnu.gp
                  sleep 2
                  show_menu;
            ;;
            10) clear;
                  option_picked "10) Find the average Prevailing Wage for
each Job for each Year (take part time and full time separate) arrange
output in descending order";
                  #echo "Enter Which Time You Want Part-Time or Full-Time"
                  #echo "For Part-Time 'N' or For Full-Time 'Y' "
                  #read time
                  time=$(yad --title "Time Selection" --entry --text '<span
foreground="red" font="14">Enter Which Time You Want Part-Time or Full-
Time\n</span><span font="12">\n<b>For Part-Time 'N' or For Full-Time 'Y'
</b>\n</span>' --width=450 --height=100 --center --button="gtk-cancel:252"
--button="gtk-ok:0")
                  #echo -e "Do you wish to see \n 1. The entire result \n2.
Year wise result\n"
                  #echo -e "choose option 1 or 2"
                  #read choice
                  choice=$(yad --title "Result Selection" --entry --text
'<span foreground="red" font="14">Do you wish to see 1. The entire result 2.
Year wise result\n</span><span font="12">\n<b>choose option 1 or 2
</b>\n</span>' --width=450 --height=100 --center --button="gtk-cancel:252"
--button="gtk-ok:0")
                  if [ $choice == 1 ];
                  then
                        rm /home/hduser/h1bproject/pig/piginput8
                        rm -r /home/hduser/h1bproject/projectout/8
                        echo -e 'time='$time'' >
/home/hduser/h1bproject/pig/piginput8
                        pig -x local -param_file
/home/hduser/h1bproject/pig/piginput8 -f
/home/hduser/h1bproject/pig/h1b8.pig | zenity --progress --title="Pig Job
Running" --pulsate --auto-close
                        echo -e "\n8) Find the average Prevailing Wage for
each Job for every Year part and full time and arrange output in descending
order\n"
                        cat /home/hduser/h1bproject/projectout/8/p*
                        sleep 5
                  else
                        #echo -e "Enter the year
(2011,2012,2013,2014,2015,2016)"
                        #read year
                        year=$(yad --title "Year Selection" --entry --text
'<span foreground="red" font="14">Enter the year
(2011,2012,2013,2014,2015,2016)\n</span><span font="12">\n<b>choose any each
year</b>\n</span>' --width=450 --height=100 --center --button="gtk-
cancel:252" --button="gtk-ok:0")
                        rm /home/hduser/h1bproject/pig/piginput8a
                        rm -r /home/hduser/h1bproject/projectout/8a
                        echo -e 'time='$time'\nyear='$year'' >
/home/hduser/h1bproject/pig/piginput8a
                        pig -x local -param_file
/home/hduser/h1bproject/pig/piginput8a -f
/home/hduser/h1bproject/pig/h1b8a.pig | zenity --progress --title="Pig Job
Running" --pulsate --auto-close
                        echo -e "\n8) Find the average Prevailing Wage for
each Job for each Year part and full time and arrange output in descending
order\n"
                        cat /home/hduser/h1bproject/projectout/8a/p*
                        sleep 5
                  fi
                  show_menu;
            ;;
            11) clear;
                  option_picked "11) Which are   employers who have the
highest success rate in petitions more than 70% in petitions and total
petions filed more than 1000?"
                  hadoop fs -rmr /niit/projout9
                  rm -r /home/hduser/h1bproject/projectout/9/*
                  hadoop jar /home/hduser/h1bproject/proj.jar
project/EmployersSuccessRate /niit/h1b/0* /niit/projout9 | zenity --progress
--title="Job Running" --pulsate --auto-close
                  hadoop fs -get /niit/projout9
/home/hduser/h1bproject/projectout/9/
                  echo -e "\n9) Which are   employers who have the highest
success rate in petitions more than 70% in petitions and total petions filed
more than 1000?\n"
                  hadoop fs -cat /niit/projout9/p*
                  sleep 5
                  show_menu;
            ;;
            12) clear;
                  option_picked "12) Which are the top 10 job positions
which have the  success rate more than 70% in petitions and total petitions
filed more than 1000?"
                  hadoop fs -rmr /niit/projout10
                  rm -r /home/hduser/h1bproject/projectout/10/*
                  hadoop jar /home/hduser/h1bproject/proj.jar
project/JobSuccessRate /niit/h1b/0* /niit/projout10 | zenity --progress
--title="Job Running" --pulsate --auto-close
                  hadoop fs -get /niit/projout10
/home/hduser/h1bproject/projectout/10/
                  echo -e "\n10) Which are the top 10 job positions which
have the  success rate more than 70% in petitions and total petitions filed
more than 1000?\n"
                  hadoop fs -cat /niit/projout10/p*
                  sleep 5
                  show_menu;
            ;;
            13) clear;
                  option_picked "11) Export result for question no 10 to
MySql database."
                  #echo "Please enter your MySql database details"
                  #read -p 'username: ' user
                  #read -sp 'password: ' password
                  user=$(yad --title "MYSQL Details" --entry --text '<span
foreground="red" font="14">Please enter your MySql database
details</span><span font="12">\n<b>Enter UserName</b>\n</span>' --width=450
--height=100 --center --button="gtk-cancel:252" --button="gtk-ok:0")
                  password=$(yad --title "MYSQL Details" --entry --text
'<span foreground="red" font="14">Please enter your MySql database
details</span><span font="12">\n<b>Enter Password</b>\n</span>' --width=450
--height=100 --center --button="gtk-cancel:252" --button="gtk-ok:0")
                  #for above mysql 5.6x set the username and password in
login-path
                  #mysql -u root -p krrish123

                  mysql_config_editor set --login-path=local
--host=localhost --user=$user --password
                  #echo -n $password > /home/hduser/import.txt
                  #hadoop fs -rm /user/import.txt
                  #hadoop fs -put /home/hduser/import.txt /user/
                  mysql --login-path=local -e "create database if not exists
project;use project;drop table if exists h1b10;create table
h1b10(employee_name varchar(100),total_application int,success_rate
varchar(40)); exit;"
                  #mysql_config_editor remove --login-path=local
                  sqoop export --connect jdbc:mysql://localhost/project
--username $user --password-file /user/import.txt --table h1b10 --update-
mode allowinsert --update-key employee_name --export-dir /niit/projout10/p*
--input-fields-terminated-by '@' ;
                  mysql --login-path=local -e "use project;select * from
h1b10;"
                  sleep 5
                  show_menu;
            ;;
            \n) exit;
            ;;
            *) clear;
            option_picked "Pick an option from the menu";
            show_menu;
            ;;
        esac
fi
done
```

Creating the table in Hive.

```
create database h1b;

use h1b;

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 bigint,year string, worksite
string, longitute double, latitute double )
ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.OpenCSVSerde'
WITH SERDEPROPERTIES (
"separatorChar" = ",",
"quoteChar" = "\""
) STORED AS TEXTFILE;

load data local inpath '/home/hduser/Downloads/Project files/h1b.csv'
overwrite into table
h1b_applications;

CREATE TABLE h1b_app2(s_no int,case_status string, employer_name
string, soc_name string, job_title string, full_time_position
string,prevailing_wage bigint,year string, worksite string, longitute
double, latitute double )
row format delimited
fields terminated by '\t'
STORED AS TEXTFILE;


INSERT OVERWRITE TABLE h1b_app2 SELECT regexp_replace(s_no, "\t", ""),
regexp_replace(case_status, "\t", ""), regexp_replace(employer_name,
"\t", ""), regexp_replace(soc_name, "\t", ""),
regexp_replace(job_title, "\t", ""),
regexp_replace(full_time_position, "\t", ""), prevailing_wage,
regexp_replace(year, "\t", ""), regexp_replace(worksite, "\t", ""),
regexp_replace(longitute, "\t", ""), regexp_replace(latitute, "\t",
"") FROM h1b_applications where case_status != "NA";

CREATE TABLE h1b_final(s_no int,case_status string, employer_name string,
soc_name string, job_title string, full_time_position string,prevailing_wage
bigint,year string, worksite string, longitute double, latitute double ) row
format delimited fields terminated by '\t' STORED AS TEXTFILE;

INSERT OVERWRITE TABLE h1b_final SELECT s_no,case when trim(case_status) =
"PENDING QUALITY AND COMPLIANCE REVIEW - UNASSIGNED" then "DENIED" when
trim(case_status) = "REJECTED" then "DENIED" when trim(case_status) =
"INVALIDATED" then "DENIED" else case_status end, employer_name, soc_name,
job_title, full_time_position, case when prevailing_wage is null then 100000
else prevailing_wage end,year, worksite, longitute, latitute FROM h1b_app2;
```

Source code and Output of all analysis

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

To analyze this we can run a MapReduce program written in java which is suitable for reducing it and perform final level operations.

Source code

//Driver Class

```
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.DoubleWritable;
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 DataEngineerJob {
public static void main(String[] args) throws Exception {

            Configuration conf = new Configuration();

            Job job = Job.getInstance(conf,"Data Engineer Job Increasing");

            job.setJarByClass(DataEngineerJob.class);

            job.setMapperClass(MyMapper.class);
            job.setReducerClass(MyReducer.class);

            job.setMapOutputKeyClass(Text.class);
            job.setMapOutputValueClass(IntWritable.class);

            job.setOutputKeyClass(Text.class);
            job.setOutputValueClass(DoubleWritable.class);

            FileInputFormat.addInputPath(job, new Path(args[0]));
            FileOutputFormat.setOutputPath(job, new Path(args[1]));

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

//Mapper class

```
public static class MyMapper extends Mapper<LongWritable, Text, Text,
IntWritable>
        {
                Text myKey = new Text();
                IntWritable one = new IntWritable(1);

                @Override
                protected void map(LongWritable key, Text value, Context
context)throws IOException, InterruptedException
                {
                        String[] record = value.toString().split("\t");

                        String job_title = record[4];

                        String year = record[7];

                        if(job_title.contains("DATA ENGINEER") && job_title != null)
                        {
                                String str = "DATA ENGINEER"+","+year;

                                myKey.set(str);

                                context.write(myKey, one);
                        }
                }

        }
```

//Reducer class

```
public static class MyReducer extends Reducer<Text, IntWritable, Text,
DoubleWritable>
        {
                String[] years = {"2011","2012","2013","2014","2015","2016"};
                double[] arr = new double[6];
                TreeMap<String,Double> map = new TreeMap<String,Double>();
                int i = 0;
                @Override
                protected void reduce(Text key, Iterable<IntWritable> values,Context
context)throws IOException, InterruptedException
                {
                        int sum =0;

                        for(IntWritable val : values)
                        {
                                sum += val.get();
                        }

                        arr[i++] = sum;

                }
                @Override
                protected void cleanup(Context context)throws IOException,
InterruptedException
                {
                   double avg = 0.0;
                   double sum1 = 0.0;
                   for(int i=0; i<6; i++ )
                   {
                         /*if(i == 0)
                         {
                                context.write(new Text(years[i]), new 
DoubleWritable(0));
                         }
                         else
                         {
                                context.write(new Text(years[i]), new 
DoubleWritable((arr[i]-arr[i-1])/arr[i-1]*100));
                         }*/
                         try {
                                sum1 += (arr[i]-arr[i-1])/arr[i-1]*100;

                         } catch (Exception e) {
                                System.out.println(e.getMessage());
                         }

                   }
                   avg = sum1 /5;
                   context.write(new Text("Data Engineer Average Growth For Five 
Years"), new DoubleWritable(avg));
             }
```

Output:

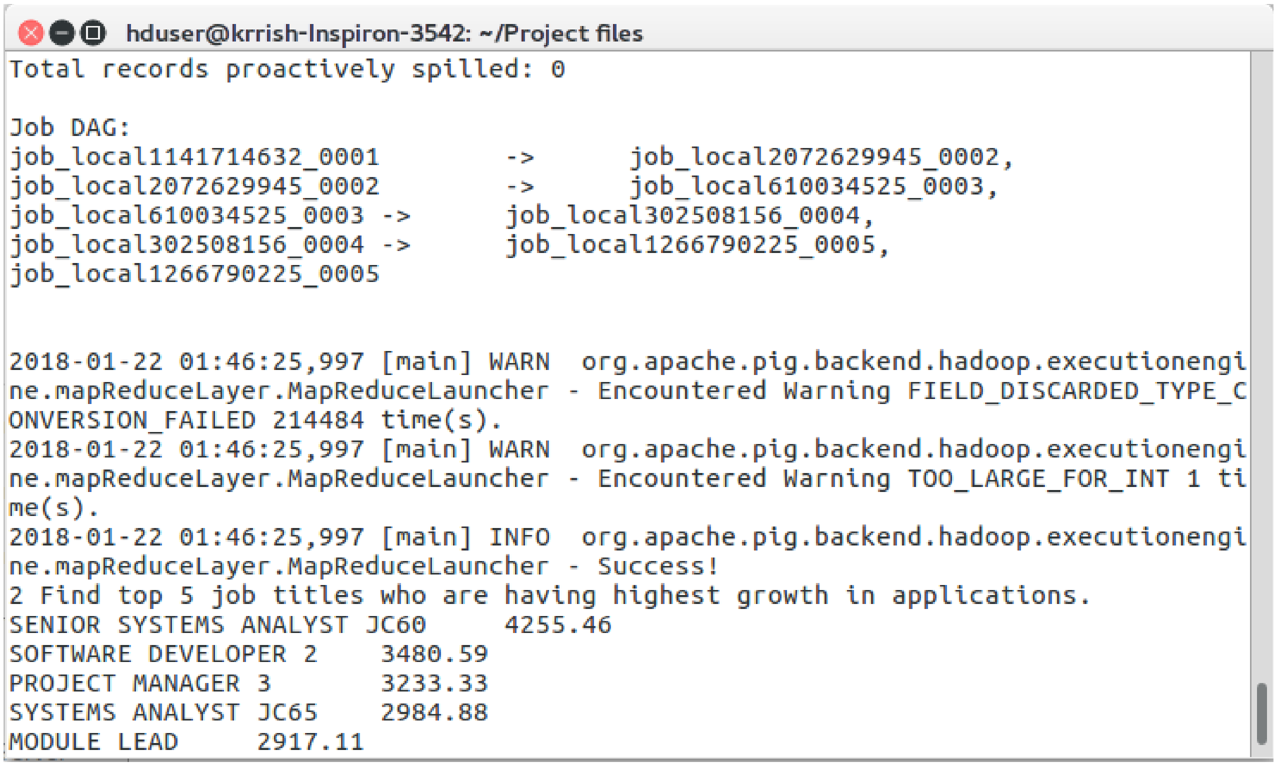
2. Find top 5 job titles who are having highest growth in applications.

This analysis is carried on Pig.

```
--1b) Find top 5 job titles who are having highest growth in applications.?

data = LOAD '/home/hduser/h1bproject/h1bdata/' 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:int,
year:chararray,
worksite:chararray,
longitute:double,
latitute:double);
noheader= filter data by $0>=1;

table1= filter noheader  by $7 matches '2011';
--dump table1;
a= group table1 by $4;
count1= foreach a generate group,COUNT($1);


table1= filter noheader  by $7 matches '2012';
```

```
a= group table1 by $4;
count2= foreach a generate group,COUNT($1);


table1= filter noheader  by $7 matches '2013';
a= group table1 by $4;
count3= foreach a generate group,COUNT($1);


table1= filter noheader  by $7 matches '2014';
a= group table1 by $4;
count4= foreach a generate group,COUNT($1);

table1= filter noheader  by $7 matches '2015';
a= group table1 by $4;
count5= foreach a generate group,COUNT($1);


table1= filter noheader  by $7 matches '2016';
a= group table1 by $4;
count6= foreach a generate group,COUNT($1);


joined= join count1 by $0,count2 by $0,count3 by $0,count4 by $0,count5 by
$0,count6 by $0;
yearwiseapplications= foreach joined generate $0,$1,$3,$5,$7,$9,$11;

--describe yearwiseapplications;
--dump yearwiseapplications;
--avg growth formula ->

growth= foreach yearwiseapplications  generate $0,
(float)($6-$5)*100/$5,(float)($5-$4)*100/$4,
(float)($4-$3)*100/$3,(float)($3-$2)*100/$2,
(float)($2-$1)*100/$1;


avggrowth= foreach growth generate $0,ROUND_TO(($1+$2+$3+$4+$5)/5,2);

orderedavggrowth= order avggrowth by $1 desc;

answer = limit orderedavggrowth 5;
--dump answer;

store answer into '/home/hduser/h1bproject/projectout/1b';

--dump answer;
```

Output:



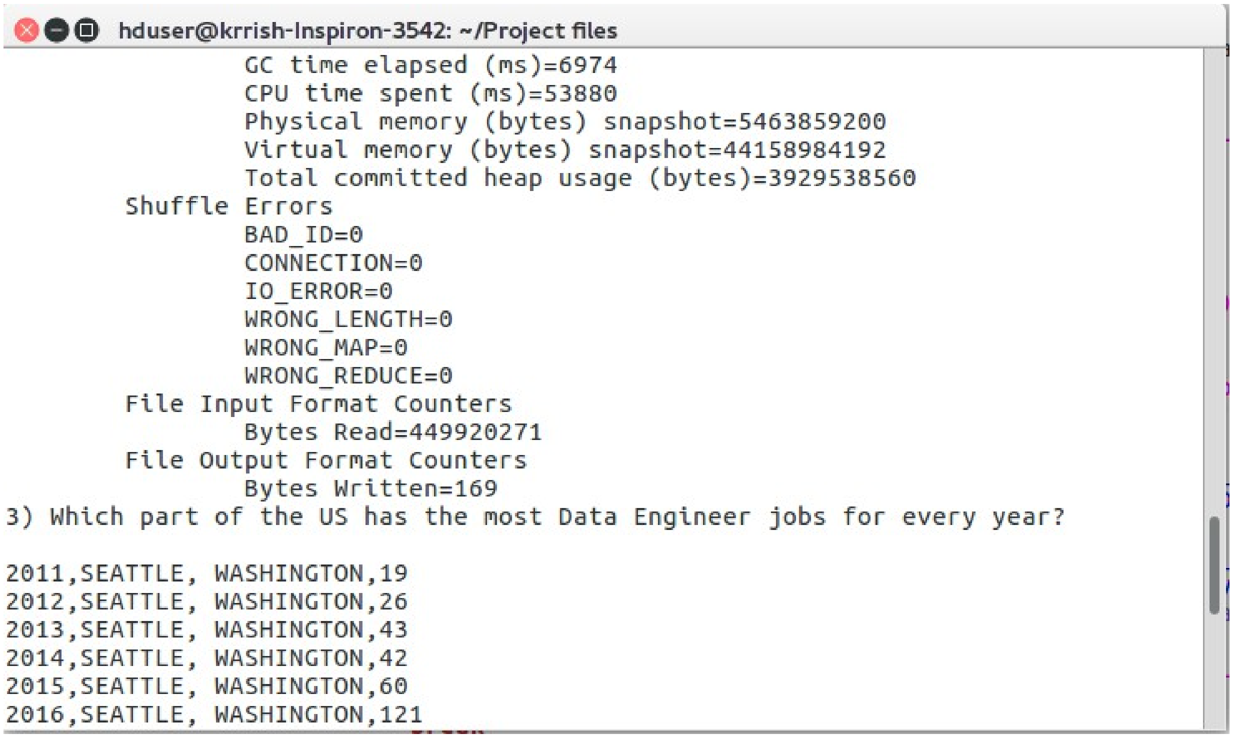
```
Total records proactively spilled: 0

Job DAG:
job_local1141714632_0001        ->      job_local2072629945_0002,
job_local2072629945_0002        ->      job_local610034525_0003,
job_local610034525_0003 ->      job_local302508156_0004,
job_local302508156_0004 ->      job_local1266790225_0005,
job_local1266790225_0005


2018-01-22 01:46:25,997 [main] WARN  org.apache.pig.backend.hadoop.executionengi
ne.mapReduceLayer.MapReduceLauncher - Encountered Warning FIELD_DISCARDED_TYPE_C
ONVERSION_FAILED 214484 time(s).
2018-01-22 01:46:25,997 [main] WARN  org.apache.pig.backend.hadoop.executionengi
ne.mapReduceLayer.MapReduceLauncher - Encountered Warning TOO_LARGE_FOR_INT 1 ti
me(s).
2018-01-22 01:46:25,997 [main] INFO  org.apache.pig.backend.hadoop.executionengi
ne.mapReduceLayer.MapReduceLauncher - Success!
2 Find top 5 job titles who are having highest growth in applications.
SENIOR SYSTEMS ANALYST JC60     4255.46
SOFTWARE DEVELOPER 2    3480.59
PROJECT MANAGER 3       3233.33
SYSTEMS ANALYST JC65    2984.88
MODULE LEAD     2917.11
```

3) Which part of the U S has the most Data Engineer jobs for each year?

```
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.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.Partitioner;
import org.apache.hadoop.mapreduce.Reducer;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;


public class WorksiteUSDataEngineerJob {

      public static class WorksiteMapper extends Mapper<LongWritable, Text,
Text, Text>
      {

            @Override
            protected void map(LongWritable key, Text value, Context
context)throws IOException, InterruptedException
            {
                  String mySearchText =
context.getConfiguration().get("myText");

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

                  String case_status = record[1];
                  String job_title = record[4];
                  String year = record[7];
                  String worksite = record[8];

                  if(mySearchText.equals("ALL"))
                  {
                        if(case_status.equals("CERTIFIED") &&
job_title.contains("DATA ENGINEER"))
                        {
                              context.write(new Text(worksite), new Text(year));
                        }
                  }

                  else{

                        if(case_status.equals("CERTIFIED") &&
job_title.contains("DATA ENGINEER") && year.equals(mySearchText))
                        {
                              context.write(new Text(worksite), new Text(year));
                        }
                  }


            }

      }

      public static class YearPartitioner extends Partitioner<Text, Text>
      {

            @Override
            public int getPartition(Text key, Text value, int numReduceTasks) {

                  String year = value.toString();

                  if(year.equals("2011"))
                  {
                        return 0 % numReduceTasks;
                  }
                  else if(year.equals("2012"))
                  {
                        return 1 % numReduceTasks;
                  }
                  else if(year.equals("2013"))
                  {
                        return 2 % numReduceTasks;
                  }
                  else if(year.equals("2014"))
                  {
```

```
                return 3 % numReduceTasks;
            }
            else if(year.equals("2015"))
            {
                return 4 % numReduceTasks;
            }
            else
            {
                return 5 % numReduceTasks;
            }
        }

    }

    public static class WorksiteReducer extends Reducer<Text, Text,
NullWritable, Text>
    {
        TreeMap<Integer, Text> map = new TreeMap<Integer,Text>();
        @Override
        protected void reduce(Text key, Iterable<Text> values,Context
context)throws IOException, InterruptedException
        {
            int count = 0;
            String year = "";
            for(Text val :values)
            {
                year = val.toString();
                count++;
            }

            String myKey = key.toString();

            String myVal = year+","+myKey+","+count;

            map.put(new Integer(count),new Text(myVal));

            if(map.size() > 1)
            {
                map.remove(map.firstKey());
            }

        }
        @Override
        protected void cleanup(Context context)throws IOException,
InterruptedException
        {
            for(Text top5 : map.descendingMap().values())
            {
                context.write(NullWritable.get(), top5);
            }
        }


    }

    public static void main(String[] args) throws Exception {
        Configuration conf  = new Configuration();

        if(args.length > 2)
        {
            conf.set("myText", args[2]);
        }

        Job job = Job.getInstance(conf, "Worsite having Most data engineer
job in US for each year");

        job.setJarByClass(WorksiteUSDataEngineerJob.class);

        job.setMapperClass(WorksiteMapper.class);

        if(args[2].equals("ALL"))
        {
            job.setPartitionerClass(YearPartitioner.class);
            job.setNumReduceTasks(6);
        }
        job.setReducerClass(WorksiteReducer.class);

        job.setMapOutputKeyClass(Text.class);
        job.setMapOutputValueClass(Text.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);
    }

}
```

Ouptut:



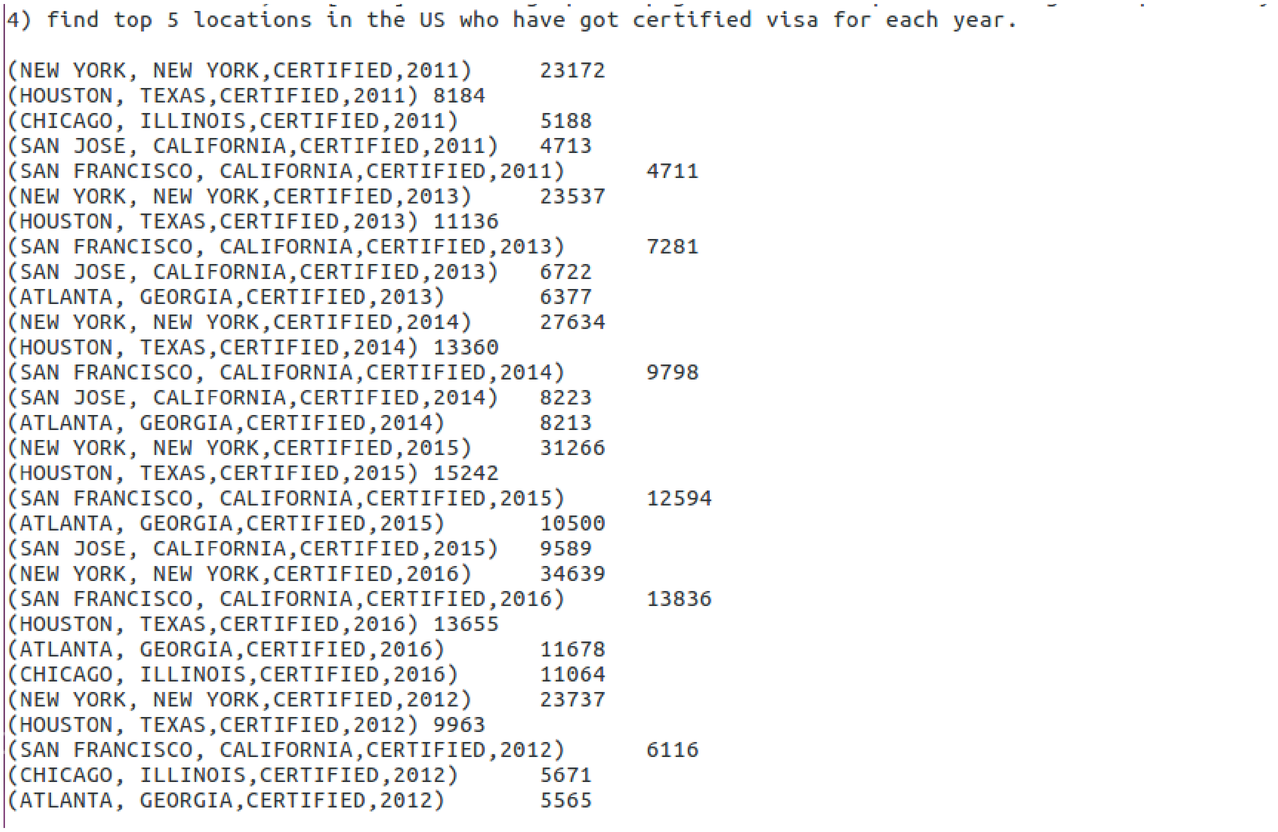
```
                GC time elapsed (ms)=6974
                CPU time spent (ms)=53880
                Physical memory (bytes) snapshot=5463859200
                Virtual memory (bytes) snapshot=44158984192
                Total committed heap usage (bytes)=3929538560
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Input Format Counters
                Bytes Read=449920271
        File Output Format Counters
                Bytes Written=169
3) Which part of the US has the most Data Engineer jobs for every year?

2011,SEATTLE, WASHINGTON,19
2012,SEATTLE, WASHINGTON,26
2013,SEATTLE, WASHINGTON,43
2014,SEATTLE, WASHINGTON,42
2015,SEATTLE, WASHINGTON,60
2016,SEATTLE, WASHINGTON,121
```

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

```
L1 = load '/home/hduser/h1bproject/h1bdata' 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: int,year:
chararray, worksite: chararray, longitude: double, latitute: double);

group1 = foreach L1 generate $8, $1, $7;
group2 = filter group1 by $1 == 'CERTIFIED';
--dump group2;


group3_2011 = filter group2 by $2 =='2011';
group3_2012 = filter group2 by $2 =='2012';
group3_2013 = filter group2 by $2 =='2013';
group3_2014 = filter group2 by $2 =='2014';
group3_2015 = filter group2 by $2 =='2015';
group3_2016 = filter group2 by $2 =='2016';


group4_2011 = group group3_2011 by ($0,$1,$2);
--dump group4_2011;
group4_2012 = group group3_2012 by ($0,$1,$2);
group4_2013 = group group3_2013 by ($0,$1,$2);
group4_2014 = group group3_2014 by ($0,$1,$2);
group4_2015 = group group3_2015 by ($0,$1,$2);
group4_2016 = group group3_2016 by ($0,$1,$2);

group5_2011 = foreach group4_2011 generate group, COUNT(group3_2011.$1);
--dump group5_2011;
group5_2012 = foreach group4_2012 generate group, COUNT(group3_2012.$1);
group5_2013 = foreach group4_2013 generate group, COUNT(group3_2013.$1);
group5_2014 = foreach group4_2014 generate group, COUNT(group3_2014.$1);
group5_2015 = foreach group4_2015 generate group, COUNT(group3_2015.$1);
```

```
group5_2016 = foreach group4_2016 generate group, COUNT(group3_2016.$1);


group_desc2011 = order group5_2011 by $1 desc;
--dump group_desc2011;
group_desc2012 = order group5_2012 by $1 desc;
group_desc2013 = order group5_2013 by $1 desc;
group_desc2014 = order group5_2014 by $1 desc;
group_desc2015 = order group5_2015 by $1 desc;
group_desc2016 = order group5_2016 by $1 desc;

group_limit1 = limit group_desc2011 5;
group_limit2 = limit group_desc2012 5;
group_limit3 = limit group_desc2013 5;
group_limit4 = limit group_desc2014 5;
group_limit5 = limit group_desc2015 5;
group_limit6 = limit group_desc2016 5;

group_ans = UNION group_limit1, group_limit2, group_limit3, group_limit4,
group_limit5, group_limit6;



store group_ans into '/home/hduser/h1bproject/projectout/2b';

--dump group_ans;
```

Output:



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

(NEW YORK, NEW YORK,CERTIFIED,2011)     23172
(HOUSTON, TEXAS,CERTIFIED,2011) 8184
(CHICAGO, ILLINOIS,CERTIFIED,2011)      5188
(SAN JOSE, CALIFORNIA,CERTIFIED,2011)   4713
(SAN FRANCISCO, CALIFORNIA,CERTIFIED,2011)      4711
(NEW YORK, NEW YORK,CERTIFIED,2013)     23537
(HOUSTON, TEXAS,CERTIFIED,2013) 11136
(SAN FRANCISCO, CALIFORNIA,CERTIFIED,2013)      7281
(SAN JOSE, CALIFORNIA,CERTIFIED,2013)   6722
(ATLANTA, GEORGIA,CERTIFIED,2013)       6377
(NEW YORK, NEW YORK,CERTIFIED,2014)     27634
(HOUSTON, TEXAS,CERTIFIED,2014) 13360
(SAN FRANCISCO, CALIFORNIA,CERTIFIED,2014)      9798
(SAN JOSE, CALIFORNIA,CERTIFIED,2014)   8223
(ATLANTA, GEORGIA,CERTIFIED,2014)       8213
(NEW YORK, NEW YORK,CERTIFIED,2015)     31266
(HOUSTON, TEXAS,CERTIFIED,2015) 15242
(SAN FRANCISCO, CALIFORNIA,CERTIFIED,2015)      12594
(ATLANTA, GEORGIA,CERTIFIED,2015)       10500
(SAN JOSE, CALIFORNIA,CERTIFIED,2015)   9589
(NEW YORK, NEW YORK,CERTIFIED,2016)     34639
(SAN FRANCISCO, CALIFORNIA,CERTIFIED,2016)      13836
(HOUSTON, TEXAS,CERTIFIED,2016) 13655
(ATLANTA, GEORGIA,CERTIFIED,2016)       11678
(CHICAGO, ILLINOIS,CERTIFIED,2016)      11064
(NEW YORK, NEW YORK,CERTIFIED,2012)     23737
(HOUSTON, TEXAS,CERTIFIED,2012) 9963
(SAN FRANCISCO, CALIFORNIA,CERTIFIED,2012)      6116
(CHICAGO, ILLINOIS,CERTIFIED,2012)      5671
(ATLANTA, GEORGIA,CERTIFIED,2012)       5565
```

5) Which industry has the most number of Data Scientist positions?

Hive -e “insert overwrite local directory '/home/hduser/h1bproject/projectout/3' row format delimited
FIELDS TERMINATED BY '\t' select soc_name,case_status ,count(soc_name) as cnt from h1b.h1b_final where job_title like '%DATA SCIENTIST%' and case_status = 'CERTIFIED' group by soc_name,case_status order by cnt desc limit 1;”

Output:

6) Which top 5 employers file the most petitions each year?

```
data = load '/home/hduser/h1bproject/h1bdata/' 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: int,year:
chararray, worksite: chararray, longitude: double, latitute: double);
noheader = filter data by $0 > 1;
data = order noheader by $0;
data = foreach data generate $1,$2,$7;
data2011 = filter data by ($2 matches '2011');
data2012 = filter data by ($2 matches '2012');
data2013 = filter data by ($2 matches '2013');
data2014 = filter data by ($2 matches '2014');
data2015 = filter data by ($2 matches '2015');
data2016 = filter data by ($2 matches '2016');

groupdata2011 = group data2011 by ($1,$2);
groupdata2012 = group data2012 by ($1,$2);
groupdata2013 = group data2013 by ($1,$2);
groupdata2014 = group data2014 by ($1,$2);
groupdata2015 = group data2015 by ($1,$2);
groupdata2016 = group data2016 by ($1,$2);


data2011 = foreach groupdata2011 generate Flatten(group),COUNT(data2011.$0);
data2012 = foreach groupdata2012 generate FLATTEN(group),COUNT(data2012.$0);
data2013 = foreach groupdata2013 generate FLATTEN(group),COUNT(data2013.$0);
data2014 = foreach groupdata2014 generate FLATTEN(group),COUNT(data2014.$0);
data2015 = foreach groupdata2015 generate FLATTEN(group),COUNT(data2015.$0);
data2016 = foreach groupdata2016 generate FLATTEN(group),COUNT(data2016.$0);

dataorderd2011 = order data2011 by $2 desc;
dataorderd2012 = order data2012 by $2 desc;
dataorderd2013 = order data2013 by $2 desc;
dataorderd2014 = order data2014 by $2 desc;
dataorderd2015 = order data2015 by $2 desc;
dataorderd2016 = order data2016 by $2 desc;

top5_2011 = limit dataorderd2011 5;
top5_2012 = limit dataorderd2012 5;
top5_2013 = limit dataorderd2013 5;
top5_2014 = limit dataorderd2014 5;
top5_2015 = limit dataorderd2015 5;
top5_2016 = limit dataorderd2016 5;

uniondata = union top5_2011,top5_2012,top5_2013,top5_2014,top5_2015,top5_2016;
uniondata = order uniondata by $1;

store uniondata into '/home/hduser/h1bproject/projectout/4';
--dump uniondata;
```

Output:

h1b.employer_name	No. of applications	h1b.year

TATA CONSULTANCY SERVICES LIMITED	5416	2011
MICROSOFT CORPORATION	4253	2011
DELOITTE CONSULTING LLP	3621	2011
WIPRO LIMITED	3028	2011
COGNIZANT TECHNOLOGY SOLUTIONS U.S. CORPORATION	2721	2011
LARSEN & TOUBRO INFOTECH LIMITED	2105	2011
INTEL CORPORATION	1952	2011
IBM CORPORATION	1822	2011
HCL AMERICA, INC.	1699	2011
DELOITTE & TOUCHE LLP	1391	2011
INFOSYS LIMITED	15818	2012
WIPRO LIMITED	7182	2012
TATA CONSULTANCY SERVICES LIMITED	6735	2012
DELOITTE CONSULTING LLP	4727	2012
IBM INDIA PRIVATE LIMITED	4074	2012
MICROSOFT CORPORATION	4067	2012
ACCENTURE LLP	2619	2012
LARSEN & TOUBRO INFOTECH LIMITED	2339	2012
ERNST & YOUNG U.S. LLP	2314	2012
HCL AMERICA, INC.	2178	2012
INFOSYS LIMITED	32223	2013
TATA CONSULTANCY SERVICES LIMITED	8790	2013
WIPRO LIMITED	6734	2013
DELOITTE CONSULTING LLP	6124	2013
ACCENTURE LLP	4994	2013
MICROSOFT CORPORATION	3902	2013
IBM INDIA PRIVATE LIMITED	3593	2013
LARSEN & TOUBRO INFOTECH LIMITED	3136	2013
HCL AMERICA, INC.	3011	2013
ERNST & YOUNG U.S. LLP	2182	2013
INFOSYS LIMITED	23759	2014
TATA CONSULTANCY SERVICES LIMITED	14098	2014
WIPRO LIMITED	8365	2014
DELOITTE CONSULTING LLP	7017	2014
ACCENTURE LLP	5498	2014

IBM INDIA PRIVATE LIMITED	5029	2014
HCL AMERICA, INC.	4749	2014
LARSEN & TOUBRO INFOTECH LIMITED	3939	2014
MICROSOFT CORPORATION	3750	2014
ERNST & YOUNG U.S. LLP	3727	2014
INFOSYS LIMITED	33245	2015
TATA CONSULTANCY SERVICES LIMITED	16553	2015
WIPRO LIMITED	12201	2015
IBM INDIA PRIVATE LIMITED	10693	2015
ACCENTURE LLP	9605	2015
DELOITTE CONSULTING LLP	7607	2015
HCL AMERICA, INC.	6110	2015
MICROSOFT CORPORATION	4575	2015
IGATE TECHNOLOGIES INC.	4554	2015
ERNST & YOUNG U.S. LLP	4144	2015
INFOSYS LIMITED	25352	2016
CAPGEMINI AMERICA INC	16725	2016
TATA CONSULTANCY SERVICES LIMITED	13134	2016
WIPRO LIMITED	10607	2016
IBM INDIA PRIVATE LIMITED	9787	2016
ACCENTURE LLP	9477	2016
DELOITTE CONSULTING LLP	7646	2016
TECH MAHINDRA (AMERICAS),INC.	6746	2016
COGNIZANT TECHNOLOGY SOLUTIONS U.S. CORPORATION	5370	2016
MICROSOFT CORPORATION	5029	2016

7) Find the most popular top 10 job positions for H1B visa applications for each year?

```
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.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.Partitioner;
import org.apache.hadoop.mapreduce.Reducer;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;


public class Top10JobPositions {

      public static class JobMapper extends Mapper<LongWritable, Text, Text,
Text>
      {

            @Override
            protected void map(LongWritable key, Text value, Context
context)throws IOException, InterruptedException
            {
                  String[] record = value.toString().split("\t");
                  String year = record[7];
                  String job_title = record[4];

                  context.write(new Text(job_title), new Text(year));
            }

      }

      public static class YearPartitioner extends Partitioner<Text, Text>
      {

            @Override
            public int getPartition(Text key, Text value, int numReduceTasks) {

                  String year = value.toString();

                  if(year.equals("2011"))
                  {
                        return 0 % numReduceTasks;
                  }
                  else if(year.equals("2012"))
                  {
                        return 1 % numReduceTasks;
                  }
                  else if(year.equals("2013"))
                  {
                        return 2 % numReduceTasks;
                  }
                  else if(year.equals("2014"))
                  {
                        return 3 % numReduceTasks;
                  }
                  else if(year.equals("2015"))
                  {
                        return 4 % numReduceTasks;
                  }
                  else
                  {
                        return 5 % numReduceTasks;
                  }
            }

      }

      public static class JobReducer extends Reducer<Text, Text, NullWritable,
Text>
      {
            TreeMap<Integer, Text> map = new TreeMap<Integer, Text>();

            @Override
            protected void reduce(Text key, Iterable<Text> values,Context
context)throws IOException, InterruptedException
            {
                  int count = 0;
                  String year = "";
                  for(Text val : values)
                  {
                         year = val.toString();
                         count++;
                  }

                  String Job_title = key.toString();
                  String myValue = year+","+Job_title +","+count;

                  map.put(new Integer(count), new Text(myValue));
                  if(map.size() > 10)
                  {
                        map.remove(map.firstKey());
                  }
            }

            @Override
            protected void cleanup( Context context)throws IOException,
InterruptedException
            {
                  for(Text top10 : map.descendingMap().values())
                  {
                        context.write(NullWritable.get(), top10);
                  }
            }



      }

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

            Configuration conf = new Configuration();

            Job job = Job.getInstance(conf, "Top 10 Job Positios for each
Year");

            job.setJarByClass(Top10JobPositions.class);

            job.setMapperClass(JobMapper.class);

            job.setPartitionerClass(YearPartitioner.class);
            job.setNumReduceTasks(6);

            job.setReducerClass(JobReducer.class);
```

```
        job.setMapOutputKeyClass(Text.class);
        job.setMapOutputValueClass(Text.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:

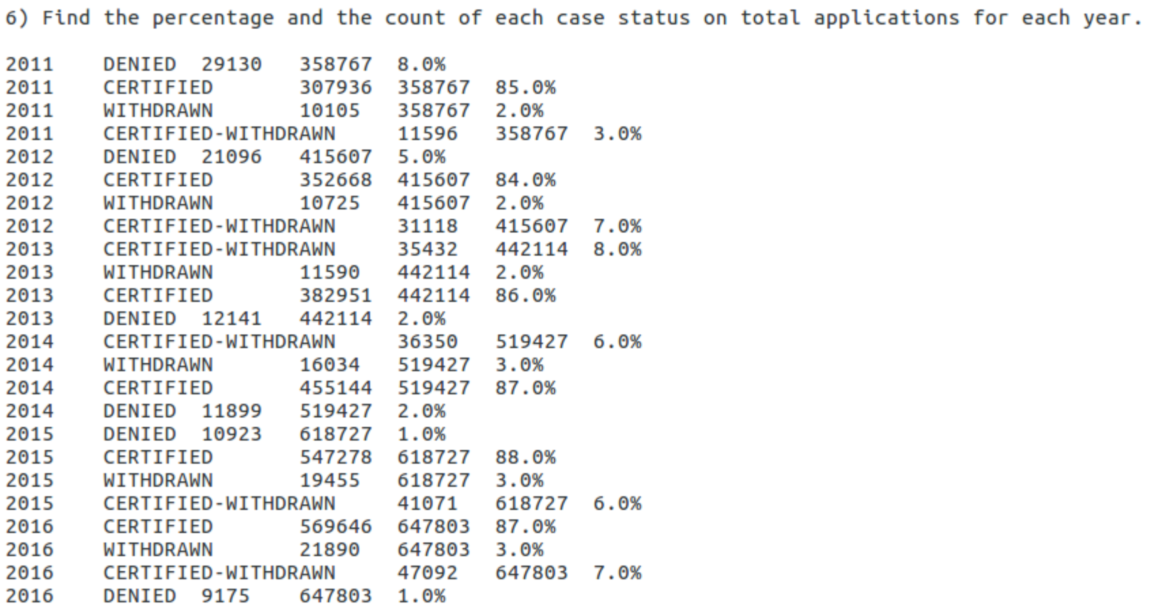
h1b.job_title	No. of applications	h1b.year
PROGRAMMER ANALYST	31799	2011
SOFTWARE ENGINEER	12763	2011
COMPUTER PROGRAMMER	8998	2011
SYSTEMS ANALYST	8644	2011
BUSINESS ANALYST	3891	2011
COMPUTER SYSTEMS ANALYST	3698	2011
ASSISTANT PROFESSOR	3467	2011
PHYSICAL THERAPIST	3377	2011
SENIOR SOFTWARE ENGINEER	2935	2011
SENIOR CONSULTANT	2798	2011
		Year = 2012
PROGRAMMER ANALYST	33066	2012
SOFTWARE ENGINEER	14437	2012
COMPUTER PROGRAMMER	9629	2012
SYSTEMS ANALYST	9296	2012
BUSINESS ANALYST	4752	2012
COMPUTER SYSTEMS ANALYST	4706	2012
SOFTWARE DEVELOPER	3895	2012
PHYSICAL THERAPIST	3871	2012
ASSISTANT PROFESSOR	3801	2012
SENIOR CONSULTANT	3737	2012
		Year = 2013
PROGRAMMER ANALYST	33880	2013
SOFTWARE ENGINEER	15680	2013
COMPUTER PROGRAMMER	11271	2013
SYSTEMS ANALYST	8714	2013
TECHNOLOGY LEAD - US	7853	2013
TECHNOLOGY ANALYST - US	7683	2013

BUSINESS ANALYST	5716	2013
COMPUTER SYSTEMS ANALYST	5043	2013
SOFTWARE DEVELOPER	5026	2013
SENIOR CONSULTANT	4326	2013
		Year = 2014
PROGRAMMER ANALYST	43114	2014
SOFTWARE ENGINEER	20500	2014
COMPUTER PROGRAMMER	14950	2014
SYSTEMS ANALYST	10194	2014
SOFTWARE DEVELOPER	7337	2014
BUSINESS ANALYST	7302	2014
COMPUTER SYSTEMS ANALYST	6821	2014
TECHNOLOGY LEAD - US	5057	2014
TECHNOLOGY ANALYST - US	4913	2014
SENIOR CONSULTANT	4898	2014
		Year = 2015
PROGRAMMER ANALYST	53436	2015
SOFTWARE ENGINEER	27259	2015
COMPUTER PROGRAMMER	14054	2015
SYSTEMS ANALYST	12803	2015
SOFTWARE DEVELOPER	10441	2015
BUSINESS ANALYST	8853	2015
TECHNOLOGY LEAD - US	8242	2015
COMPUTER SYSTEMS ANALYST	7918	2015
TECHNOLOGY ANALYST - US	7014	2015
SENIOR SOFTWARE ENGINEER	6013	2015
		Year = 2016
PROGRAMMER ANALYST	53743	2016
SOFTWARE ENGINEER	30668	2016
SOFTWARE DEVELOPER	14041	2016
SYSTEMS ANALYST	12314	2016
COMPUTER PROGRAMMER	11668	2016
BUSINESS ANALYST	9167	2016
COMPUTER SYSTEMS ANALYST	6900	2016
SENIOR SOFTWARE ENGINEER	6439	2016
DEVELOPER	6084	2016
TECHNOLOGY LEAD - US	5410	2016

8. 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.

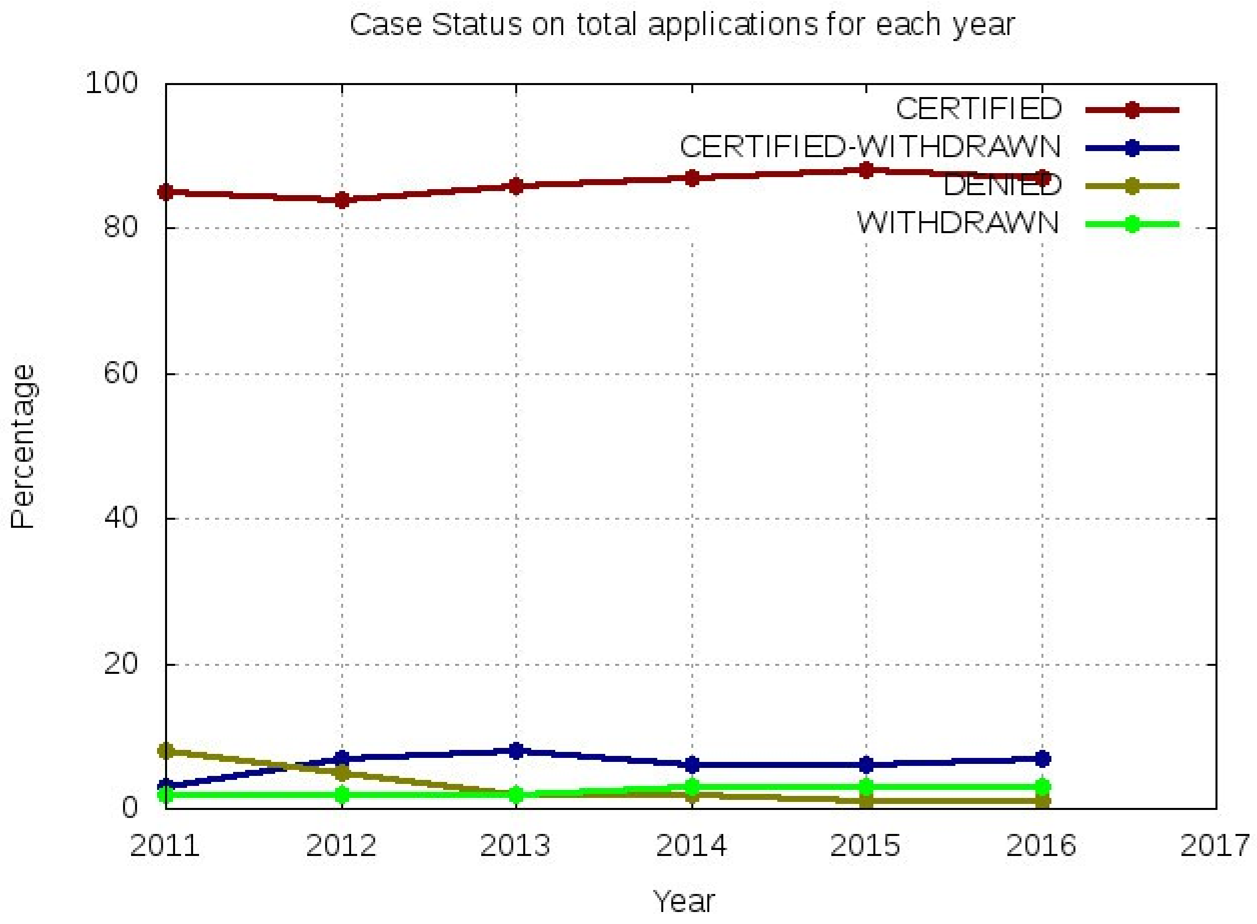
```
table1 = load '/home/hduser/h1bproject/h1bdata' using PigStorage('\t') as
(s_no,case_status,employer_name,soc_name,job_title,full_time_position
,prevailing_wage,year,worksite,longitute,latitute);
noheader = filter table1 by $0 > '0' ;
table2 = order noheader by $0;
table3 = group table2 by (year);
table4 = FOREACH table3 GENERATE FLATTEN(group) AS year,COUNT(table2.case_status) as
total_case_status;

table5 = group table2 by (year,case_status);
--dump table5;
table6 = FOREACH table5 GENERATE
   FLATTEN(group) AS (year,case_status),COUNT(table2.case_status) as total_case_status;
join_table = join table6 by year, table4 by year;
table7 = foreach join_table generate $0,$1,$2,$4;

table8 = foreach table7 generate  $0,$1,$2,$3,CONCAT((chararray)ROUND_TO((float)(($2*100)/
$3),2),'%');

--describe table8;
filtcer = filter table8 by ($1 matches 'CERTIFIED');
filtden = filter table8 by ($1 matches 'DENIED');
filtcerwith = filter table8 by ($1 matches 'CERTIFIED-WITHDRAWN');
filtwith = filter table8 by ($1 matches 'WITHDRAWN');
--dump filtyr2011;

store filtcer into '/home/hduser/h1bproject/graph/6/data/filtcer';
store filtden into '/home/hduser/h1bproject/graph/6/data/filtden';
store filtcerwith into '/home/hduser/h1bproject/graph/6/data/filtcerwith';
store filtwith into '/home/hduser/h1bproject/graph/6/data/filtwith';


store table8 into '/home/hduser/h1bproject/projectout/6';

--dump table8;
```

Output:



```
6) Find the percentage and the count of each case status on total applications for each year.

2011    DENIED  29130   358767  8.0%
2011    CERTIFIED       307936  358767  85.0%
2011    WITHDRAWN       10105   358767  2.0%
2011    CERTIFIED-WITHDRAWN     11596   358767  3.0%
2012    DENIED  21096   415607  5.0%
2012    CERTIFIED       352668  415607  84.0%
2012    WITHDRAWN       10725   415607  2.0%
2012    CERTIFIED-WITHDRAWN     31118   415607  7.0%
2013    CERTIFIED-WITHDRAWN     35432   442114  8.0%
2013    WITHDRAWN       11590   442114  2.0%
2013    CERTIFIED       382951  442114  86.0%
2013    DENIED  12141   442114  2.0%
2014    CERTIFIED-WITHDRAWN     36350   519427  6.0%
2014    WITHDRAWN       16034   519427  3.0%
2014    CERTIFIED       455144  519427  87.0%
2014    DENIED  11899   519427  2.0%
2015    DENIED  10923   618727  1.0%
2015    CERTIFIED       547278  618727  88.0%
2015    WITHDRAWN       19455   618727  3.0%
2015    CERTIFIED-WITHDRAWN     41071   618727  6.0%
2016    CERTIFIED       569646  647803  87.0%
2016    WITHDRAWN       21890   647803  3.0%
2016    CERTIFIED-WITHDRAWN     47092   647803  7.0%
2016    DENIED  9175    647803  1.0%
```

Graph :



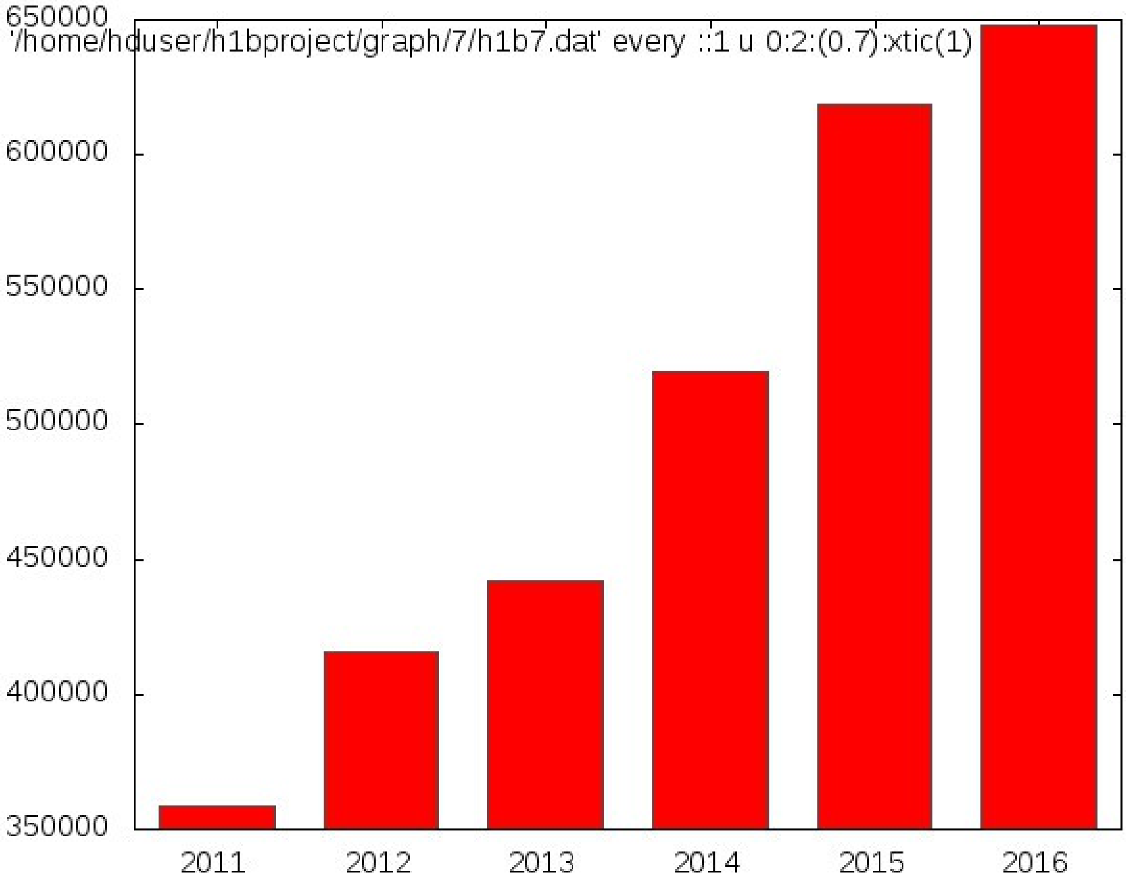
9) Create a bar graph to depict the number of applications for each year.

```
use h1b;

INSERT OVERWRITE LOCAL DIRECTORY '/home/hduser/h1bproject/graph/h1b7.dat'
select a.year,count(a.year) as no_of_applications from h1b_final a where a.year
is not NULL group by a.year order by a.year;
```

```
Output:
```

h1b.year	No. of applications
2011	358767
2012	415605
2013	442110
2014	519426
2015	618727
2016	647803

Graph :


650000
600000
550000
500000
450000
400000
350000
'/home/hduser/h1bproject/graph/7/h1b7.dat' every ::1 u 0:2:(0.7):xtic(1)
2011
2012
2013
2014
2015
2016

10) Find the average Prevailing Wage for each Job for each Year (take part time and full time separate).Arrange the output in descending order.

```
REGISTER piggybank.jar;
DEFINE CSVExcelStorage org.apache.pig.piggybank.storage.CSVExcelStorage();
h1b = load '/project/h1b/h1b.csv' using CSVExcelStorage(',') as
(s_no:int,case_status:chararray, employer_name:chararray,
soc_name:chararray, job_title:chararray,
full_time_position:chararray,prevailing_wage:int,year:chararray,
worksite:chararray, longitute:double, latitute:double);

h1b1 = filter h1b by $1 != 'CASE_STATUS';
h1b2 = filter h1b1 by $1 == 'CERTIFIED';

-- for fulltime
h1b3 = filter h1b2 by $5 == 'Y';
h1b_required = foreach h1b2 generate $8,$7,$6;
h1b_2011 = filter h1b_required by $1=='2011';
h1b_2012 = filter h1b_required by $1=='2012';
h1b_2013 = filter h1b_required by $1=='2013';
h1b_2014 = filter h1b_required by $1=='2014';
h1b_2015 = filter h1b_required by $1=='2015';
h1b_2016 = filter h1b_required by $1=='2016';


h1b_group1  =  group  h1b_2011  by  ($0,$1);
h1b_group2  =  group  h1b_2012  by  ($0,$1);
h1b_group3  =  group  h1b_2013  by  ($0,$1);
h1b_group4  =  group  h1b_2014  by  ($0,$1);
h1b_group5  =  group  h1b_2015  by  ($0,$1);
h1b_group6 = group h1b_2016 by ($0,$1);

h1b_count1  =  foreach  h1b_group1  generate  ROUND_TO(AVG(h1b_2011.$2),2),group;
h1b_count2  =  foreach  h1b_group2  generate  ROUND_TO(AVG(h1b_2012.$2),2),group;
h1b_count3  =  foreach  h1b_group3  generate  ROUND_TO(AVG(h1b_2013.$2),2),group;
h1b_count4  =  foreach  h1b_group4  generate  ROUND_TO(AVG(h1b_2014.$2),2),group;
h1b_count5  =  foreach  h1b_group5  generate  ROUND_TO(AVG(h1b_2015.$2),2),group;
h1b_count6 = foreach h1b_group6 generate ROUND_TO(AVG(h1b_2016.$2),2),group;

h1b_union_y = UNION
h1b_count1,h1b_count2,h1b_count3,h1b_count4,h1b_count5,h1b_count6;
h1b_order_y = order h1b_union_y by $0 desc;

-- for parttime

h1b3 = filter h1b2 by $5 == 'N';
h1b_required = foreach h1b2 generate $8,$7,$6;
h1b_2011 = filter h1b_required by $1=='2011';
```

```
h1b_2012 = filter h1b_required by $1=='2012';
h1b_2013 = filter h1b_required by $1=='2013';
h1b_2014 = filter h1b_required by $1=='2014';
h1b_2015 = filter h1b_required by $1=='2015';
h1b_2016 = filter h1b_required by $1=='2016';


h1b_group1  =  group  h1b_2011  by  ($0,$1);
h1b_group2  =  group  h1b_2012  by  ($0,$1);
h1b_group3  =  group  h1b_2013  by  ($0,$1);
h1b_group4  =  group  h1b_2014  by  ($0,$1);
h1b_group5  =  group  h1b_2015  by  ($0,$1);
h1b_group6 = group h1b_2016 by ($0,$1);

h1b_count1  =  foreach  h1b_group1  generate  ROUND_TO(AVG(h1b_2011.$2),2),group;
h1b_count2  =  foreach  h1b_group2  generate  ROUND_TO(AVG(h1b_2012.$2),2),group;
h1b_count3  =  foreach  h1b_group3  generate  ROUND_TO(AVG(h1b_2013.$2),2),group;
h1b_count4  =  foreach  h1b_group4  generate  ROUND_TO(AVG(h1b_2014.$2),2),group;
h1b_count5  =  foreach  h1b_group5  generate  ROUND_TO(AVG(h1b_2015.$2),2),group;
h1b_count6 = foreach h1b_group6 generate ROUND_TO(AVG(h1b_2016.$2),2),group;

h1b_union_n = UNION
h1b_count1,h1b_count2,h1b_count3,h1b_count4,h1b_count5,h1b_count6;
h1b_order_n = order h1b_union_n by $0 desc;

---union of those

h1b_union = union h1b_union_y,h1b_union_n;

store h1b_union into '/project/h1b/analysis10' using PigStorage(',');
```

Sample Output (complete output can be found here):

Average Salary	Job Title	Year	Full Time(Y or N)
67267	DB2 DBA	2016	N
74734	DB2 DBA	2016	Y
102606	DENITST	2016	Y
58905.62	DENTIST	2016	N
111861.4	DENTIST	2016	Y
100817	DOR-OTR	2016	Y
41520.8	DRAFTER	2016	N
65270	EDITORS	2016	N
78354	ENGINER	2016	Y
51036	FACULTY	2016	N
87130.5	FACULTY	2016	Y
53706	IT LEAD	2016	N

89107	IT LEAD	2016	Y
53789	JEWELER	2016	N
23316	LABORER	2016	N
34258	LANDMAN	2016	N
84951.2	LAWYERS	2016	Y
63623	LEAD-QA	2016	N
71115	LEAD-QA	2016	Y
60224.88	MANAGER	2016	N
98056.83	MANAGER	2016	Y
94557	MANGAER	2016	Y
60153.55	MODELER	2016	N
81349	MODELER	2016	Y

11) Which are employers along with the number of petitions who have the success rate more than 70% in petitions and total petitions filed more than 1000?

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

public class JobSuccessRate
{

        public static class MapperClass extends Mapper <LongWritable, Text, Text, Text>
        {

                @Override
                public void map(LongWritable key, Text value, Context context) throws IOException,
InterruptedException
                {
                        String parts[] =value.toString().split("\t");
                        String status = parts[1];
                        String jobposition =parts[4].replaceAll("\"", "");
                        context.write(new Text(Employer), new Text(status));
                }
```

```
    }


    public static class ReducerClass extends Reducer <Text,Text,NullWritable,Text>
    {

        private TreeMap<Double, String> topten = new TreeMap<>();
        public void reduce(Text key, Iterable<Text> value, Context context){
             double total =0;
             double successrate=0;
             for (Text val:value)
             {
                 String status = val.toString();
                 if(status.equals("CERTIFIED") || status.equals("CERTIFIED
WITHDRAWN"))
                 {
                     total++ ;
                   successrate++;
                 }
                   else
                     total++;
                     }
            double rate = (successrate/total)*100;
            if(rate >=70 && total >=1000){
                String op = key.toString()+ "@"+String.format("%.0f",total)+"@" +
String.format("%.2f %%",rate);

            topten.put(rate, op);
            }


        }
        protected void cleanup(Context context) throws IOException, InterruptedException{
            for(String val : topten.values()){
                context.write(NullWritable.get(),new Text(val));
            }


        }

    }

    public static void main(String[] args ) throws Exception
    {
        Configuration conf =new Configuration();
        conf.set("mapreduce.output.textoutputformat.separator", ",");
        Job job=Job.getInstance(conf);
        job.setJarByClass(JobSuccessRate.class);

        job.setMapperClass(MapperClass.class);
```

```
                job.setReducerClass(ReducerClass.class);
                job.setOutputKeyClass(NullWritable.class);
                job.setOutputValueClass(Text.class);
                job.setMapOutputKeyClass(Text.class);
                job.setMapOutputValueClass(Text.class);

                FileInputFormat.addInputPath(job, new Path(args[0]));
                FileOutputFormat.setOutputPath(job,new Path(args[1]));
                System.exit(job.waitForCompletion(true) ? 0 : 1);


        }

}
```

Sample Output :

Company name	Total applications	Success Rate
HTC GLOBAL SERVICES, INC.	1164	100.00%
INFOSYS LIMITED	130592	99.54%
DIASPARK, INC.	1419	99.51%
ACCENTURE LLP	33447	99.39%
TECH MAHINDRA (AMERICAS),INC.	10732	99.34%
TATA CONSULTANCY SERVICES LIMITED	64726	99.34%
YASH TECHNOLOGIES, INC.	2214	99.28%
YASH & LUJAN CONSULTING, INC.	1372	99.27%
HCL AMERICA, INC.	22678	99.27%
RELIABLE SOFTWARE RESOURCES, INC.	1992	99.15%

```
    }
```

11) Which are the job positions along with the number of petitions which have the success rate more than 70% in petitions and total petitions filed more than 1000?

```
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.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.FileInputFor
mat;
import
org.apache.hadoop.mapreduce.lib.output.FileOutputF
ormat;
import org.apache.hadoop.io.NullWritable;

public class JobSuccessRate
{
        public static class MapperClass extends Mapper <LongWritable, Text, Text,
Text>
        {
                @Override
                public void map(LongWritable key, Text value, Context context)
throws IOException, InterruptedException
                {
                        String parts[] =value.toString().split("\t");
                        String status = parts[1];
                        String jobposition =parts[4].replaceAll("\"", "");
                        context.write(new Text(jobposition), new Text(status));
                }

        }
```

```
    public static class ReducerClass extends Reducer
<Text,Text,NullWritable,Text>
    {

        private TreeMap<Double, String> topten = new TreeMap<>();
        public void reduce(Text key, Iterable<Text> value, Context context){
            double total =0;
            double successrate=0;
            for (Text val:value)
            {
                String status = val.toString();
                      if(status.equals("CERTIFIED") ||
status.equals("CERTIFIED WITHDRAWN"))
                {
                    total++ ;
                 successrate++;
                }
                 else
                    total++;
                    }
            double rate = (successrate/total)*100;
            if(rate >=70 && total >=1000){
                String op = key.toString()+
"@"+String.format("%.0f",total)+"@" + String.format("%.2f %%",rate);

            topten.put(rate, op);
            }


        }
        protected void cleanup(Context context) throws IOException,
InterruptedException{
            for(String val : topten.values()){
                context.write(NullWritable.get(),new Text(val));
            }


        }
    }

    public static void main(String[] args ) throws Exception
    {
        Configuration conf =new Configuration();
        conf.set("mapreduce.output.textoutputformat.separator", ",");
        Job job=Job.getInstance(conf);
        job.setJarByClass(JobSuccessRate.class);

        job.setMapperClass(MapperClass.class);


        job.setReducerClass(ReducerClass.class);
        job.setOutputKeyClass(NullWritable.class);
        job.setOutputValueClass(Text.class);
        job.setMapOutputKeyClass(Text.class);
        job.setMapOutputValueClass(Text.class);

        FileInputFormat.addInputPath(job, new Path(args[0]));
        FileOutputFormat.setOutputPath(job,new Path(args[1]));
        System.exit(job.waitForCompletion(true) ? 0 : 1);


    }

}
```

Sample Output :

Job Possitions	Applications Count	Success Rate
PRODUCTION SUPPORT LEAD - US	1301	100.00%
ASSOCIATE CONSULTANT - US	4393	99.93%
SYSTEMS ENGINEER - US	10036	99.90%
TEST ENGINEER - US	2198	99.86%
PRODUCTION SUPPORT ANALYST - US	1451	99.86%
TEST ANALYST - US	4958	99.82%
CONSULTANT - US	7426	99.81%
TECHNOLOGY LEAD - US	28350	99.80%
TECHNICAL TEST LEAD - US	5374	99.80%
SENIOR TECHNOLOGY ARCHITECT - US	1417	99.79%

12) Export result for question no 10 to MySql database.

```
#echo "Please enter your MySql database details"
                                #read -p 'username: ' user
                                #read -sp 'password: ' password
                                user=$(yad --title "MYSQL
Details" --entry --text '<span foreground="red"
font="14">Please enter your MySql database
details</span><span font="12">\n<b>Enter
UserName</b>\n</span>' --width=450 --height=100
--center --button="gtk-cancel:252" --button="gtk-
ok:0")
                                password=$(yad --title
"MYSQL Details" --entry --text '<span
foreground="red" font="14">Please enter your MySql
database details</span><span font="12">\n<b>Enter
Password</b>\n</span>' --width=450 --height=100
--center --button="gtk-cancel:252" --button="gtk-
ok:0")
                                #for above mysql 5.6x set the
username and password in login-path
                                #mysql -u root -p krrish123

                                mysql_config_editor set --login-
path=local --host=localhost --user=$user --password
                                #echo -n $password >
/home/hduser/import.txt
                                #hadoop fs -rm /user/import.txt
                                #hadoop fs -put
/home/hduser/import.txt /user/
                                mysql --login-path=local -e
"create database if not exists project;use project;drop
table if exists h1b10;create table
h1b10(employee_name varchar(100),total_application
int,success_rate varchar(40)); exit;"
                                #mysql_config_editor remove
--login-path=local
                                sqoop export --connect
jdbc:mysql://localhost/project --username $user
--password-file /user/import.txt --table h1b10 --update-
mode allowinsert --update-key employee_name
--export-dir /niit/projout10/p* --input-fields-
```

```
terminated-by '@' ;
                              mysql --login-path=local -e "use
project;select * from h1b10;"
                              sleep 5
```

CONCLUSION

The H-1B visa has been the most popular long-term work visa in the United States for years, and with good reason. There is a whole host of benefits that give the H1B an edge over the other work visa categories. From its accessibility to its lengthy initial period of stay, it’s easy to see why so many foreign professionals apply to reap the H1B visa benefits each year. The first H1B visa benefit, and perhaps the main reason for its popularity, is the broad requirements associated with qualifying for this visa. Another benefit of the H1B visa is the amount of time you are initially granted when you receive your visa. In contrast to some of the other visas such as B1, which grants you six months, and the J-1, which can sometimes grant you as little as one year, the H1B allows holders to stay for three years initially and can easily be extended. One of the biggest H-1B benefits is that foreign professionals from all over the world can apply. While the E2 visa can only be obtained by people from treaty countries and the TN is reserved for Canadians and Mexicans, the H-1B is open to nationals and citizens of any country. Unlike many other work visa classifications, one of the H-1B's many benefits is the fact that it is considered to be a "dual intent" visa**.** This means that you can pursue legal permanent residency while under H1B nonimmigrant status. This is a large advantage over some other visas such as the TN and J1 classifications.

GitHub Link : - https://github.com/kishan9886767771/H1-B-BigData-Project