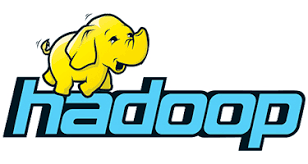
Big Data Analysis in Hadoop on H1B Visa Applicants Data



Presented By:

Shashank Akhouri

S171107200459

**Table of Content**

1. Abstract…………………………………………………………………………………3
2. Why is Hadoop Important……………………………………………………………....4
3. Hadoop Glossary………………………………………………………………………..5
4. Hadoop Ecosystem Components Used In This Project…………………………………6
5. Acknowledgements……………………………………………………………………..7
6. Project Outline…………………………………………………………………………..8
7. Project Implementation………………………………………………………………….9
8. Implementation steps……………………………………………………………………12
9. Conclusion……………………………………………………………………………....26
10. Webography…………………………………………………………………………….27

**Abstract**

**Big data**

Big data is data sets that are so voluminous and complex that traditional data-processing application software are inadequate to deal with them. Big data challenges include capturing data, data storage, data analysis, search, sharing, transfer, visualization, querying, updating, information privacy and data source.

**Hadoop**

Hadoop is an open source, Java-based programming framework that supports the processing and storage of extremely large data sets in a distributed computing environment. It is part of the Apache project sponsored by the Apache Software Foundation. It is an open source software framework for storing data and running applications on cluster commodity hardware. It provides massive storage for any kind of data, enormous processing power and the ability to handle virtually limitless concurrent tasks or jobs.

**Hadoop history**

* Hadoop was created by Doug Cutting who had created the Apache Lucene (Text Search), which is origin in Apache Nutch (Open source search Engine).Hadoop is a part of Apache Lucene Project. Actually Apache Nutch was started in 2002 for working crawler and search system.Nutch Architecture would not scale up to billions of pages on the web.
* In 2003 Google had published one Architecture called Google Distributed File system(GFS),which was solve the storage need for the very large files generated as a part of the web crawl and indexing process.
* In 2004 based on GFS architecture Nutch was implementing open source called the Nutch Distributed File system (NDFS).In 2004 Google was published Mapreduce,In 2005 Nutch developers had working on Mapreduce in Nutch Project. Most of the Algorithms had been ported to run using mapreduce and NDFS.
* In February 2006 they moved out of Nutch to form an independent subproject of Lucene called Hadoop. At around the same time, Doug Cutting joined Yahoo!, which provided a dedicated team and the resources to turn Hadoop into a system that ran at web scale. This was demonstrated in February 2008 when Yahoo! announced that its production search index was being generated by a 10,000-core Hadoop cluster.
* In January 2008, Hadoop was made its own top-level project at Apache, confirming its success and its diverse, active community. By this time, Hadoop was being used by many other companies besides Yahoo!, such as Last.fm, Facebook, and the New York Times.
* In April 2008, Hadoop broke a world record to become the fastest system to sort a terabyte of data. Running on a 910-node cluster, Hadoop sorted one terabyte in 209 seconds (just under 3½ minutes), beating the previous year’s winner of 297 seconds.

**Why is Hadoop Important?**

* **Storage and Processing speed**. With data volumes and varieties constantly increasing, especially from social media and the Internet of Things (IoT), that's a key consideration.
* **Computing power**. Hadoop's distributed computing model processes big data fast. The more computing nodes you use the more processing power you have.
* **Fault tolerance**. Data and application processing are protected against hardware failure. If a node goes down, jobs are automatically redirected to other nodes to make sure the distributed computing does not fail. Multiple copies of all data are stored automatically.
* **Flexibility**. Unlike traditional relational databases, you don’t have to preprocess data before storing it. You can store as much data as you want and decide how to use it later. That includes unstructured data like text, images and videos.
* **Low cost**. The open-source framework is free and uses commodity hardware to store large quantities of data.
* **Scalability**. You can easily grow your system to handle more data simply by adding nodes. Little administration is required.

**Hadoop Glossary**

* **Hadoop Common:** Usually only referred to by programmers, Hadoop Common is a common utilities library that contains code to support some of the other modules within the Hadoop ecosystem. When Hive and HBase want to access HDFS, for example, they do so using JARs (Java archives), which are libraries of Java code stored in Hadoop Common.
* **HBase:** An open-source, distributed, versioned, non-relational database modeled after Google’s
* **YARN:** A resource manager for Hadoop 2. YARN is short for “Yet another resource negotiator”.
* **MapReduce:** A parallel processing software framework. It is comprised of two steps. Map step is a master node that takes input and partitioned them into smaller sub-problems and then distributed them to worker nodes. After the map step has taken place, the master node takes the answers to all the sub problems and combines them to produce output.

**Hadoop Ecosystem Components Used In This Project**

**Pig:** Apache Pig is a high level language built on top of MapReduce for analyzing large datasets with simple adhoc data analysis programs. Pig is also known as Data Flow language. It is very well integrated with python. It is initially developed by yahoo.

Salient features of pig:

* Ease of programming
* Optimization opportunities
* Extensibility

Pig scripts internally will be converted to map reduce programs.

**Hive:** Apache Hive is another high level query language and data warehouse infrastructure built on top of Hadoop for providing data summarization, query and analysis. It is initially developed by yahoo and made open source.

Salient features of hive:

* SQL like query language called HQL.
* Partitioning and bucketing for faster data processing.
* Integration with visualization tools like Tableau.

Hive queries internally will be converted to map reduce programs.

**Apache Sqoop:** Apache Sqoop is a tool designed for bulk data transfers between relational databases and Hadoop.

Features:

* Import and export to and from HDFS.
* Import and export to and from Hive.
* Import and export to HBase.

**Apache Zookeeper:** Zookeeper is a centralized service designed for maintaining configuration information, naming, providing distributed synchronization, and providing group services.

Features:

* Serialization
* Atomicity
* Reliability
* Simple API

**Acknowledgements**

I wish to thank our Master Trainer Mr. Sandeep Aggarwal for providing complete learning on Big Data and Hadoop and guiding us in accomplishing the objective of our project, Mr. Annu Sharma for providing base training to understand programming and our Tech Mentor Mr. Naveen Goyal for guiding us at every step to understand the syllabus and building confidence in us to complete the project.

**Project Outline**

|  |  |
| --- | --- |
| **Title** | Big Data Analysis in Hadoop on H1B Visa Applicants Data |
| **Inputs** | H1B Visa Applicant Data |
| **Data Elements** | 1. **CASE\_STATUS:** Status associated with the last significant event or decision. Valid values include “Certified,” “Certified-Withdrawn,” Denied,” and “Withdrawn”. |
| * + 1. Certified: Employer filed the LCA, which was approved by DOL |
| * + 1. Certified Withdrawn: LCA was approved but later withdrawn by employer |
| * + 1. Withdrawn: LCA was withdrawn by employer before approval |
| * + 1. Denied: LCA was denied by DOL |
| 1. **EMPLOYER\_NAME:** Name of employer submitting labour condition application. |
| 1. **SOC\_NAME:** the Occupational name associated with the SOC\_CODE. SOC\_CODE is the occupational code associated with the job being requested for temporary labour condition, as classified by the Standard Occupational Classification (SOC) System. |
| 1. **JOB\_TITLE:** Title of the job |
| 1. **FULL\_TIME\_POSITION:** Y = Full Time Position; N = Part Time Position |
| 1. **PREVAILING\_WAGE:** Prevailing Wage for the job being requested for temporary labour 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. |
| 1. **YEAR:** Year in which the H1B visa petition was filed |
| 1. **WORKSITE:** City and State information of the foreign worker’s intended area of employment |
| 1. **lon:** longitude of the Worksite |
| 1. **lat:** latitude of the Worksite |
| **Analysis Relevance** | 1. a) Is the number of petitions with Data Engineer job title increasing over time? |
| b) Find top 5 job titles who are having highest avg growth in applications.[ALL] |
| 1. a) Which part of the US has the most Data Engineer jobs for each year? |
| b) find top 5 locations in the US who have got certified visa for each year.[certified] |
| 1. Which industry (SOC\_NAME) has the most number of Data Scientist positions? [certified] |
| 1. Which top 5 employers file the most petitions each year? - Case Status - ALL |
| 1. Find the most popular top 10 job positions for H1B visa applications for each year? |
| 1. For all the applications |
| 1. For only certified applications. |
| 1. Find the percentage and the count of each case status on total applications for each year. Create a line graph depicting the pattern of All the cases over the period of time. |
| 1. Create a bar graph to depict the number of applications for each year [All] |
| 1. Find the average Prevailing Wage for each Job for each Year (take part time and full time separate). Arrange the output in descending order - [Certified and Certified Withdrawn.] |
| 1. Which are the employers along with the number of petitions who have the success rate more than 70% in petitions. (total petitions filed 1000 OR more than 1000) ? |
| 1. Which are the job positions along with the number of petitions which have the success rate more than 70% in petitions (total petitions filed 1000 OR more than 1000)? |
| 1. Export result for question no 10 to MySql database. |
| **Purpose** | We will be performing analysis on the H1B visa applicants between the years 2011-2016. After analyzing the data, we can derive set of facts. Frequently Asked Questions to analyze different trends of applicants |
| **Methodology** | Agile |

**Project Implementation**

**Assumption:**

1. Hadoop cluster is running
2. Ecosystem products (Pig, Hive, Sqoop) are installed
3. MYSQL is installed
4. H1B data is available on HDFS and local machine

**Prerequisites for all jobs:**

The dataset has nearly 3 million records.

In the data, few columns are enclosed by double quotes and also we have commas in a single column and the column is enclosed by double quotes. So we have used hive csv serve to load the data. In the quote Char, we have given **“(**double quote**).**So this will take the column value in between the double quotes.

Steps given below to clean the data:

Let’s create a table to load the h1b applicant’s data as shown below.

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/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";

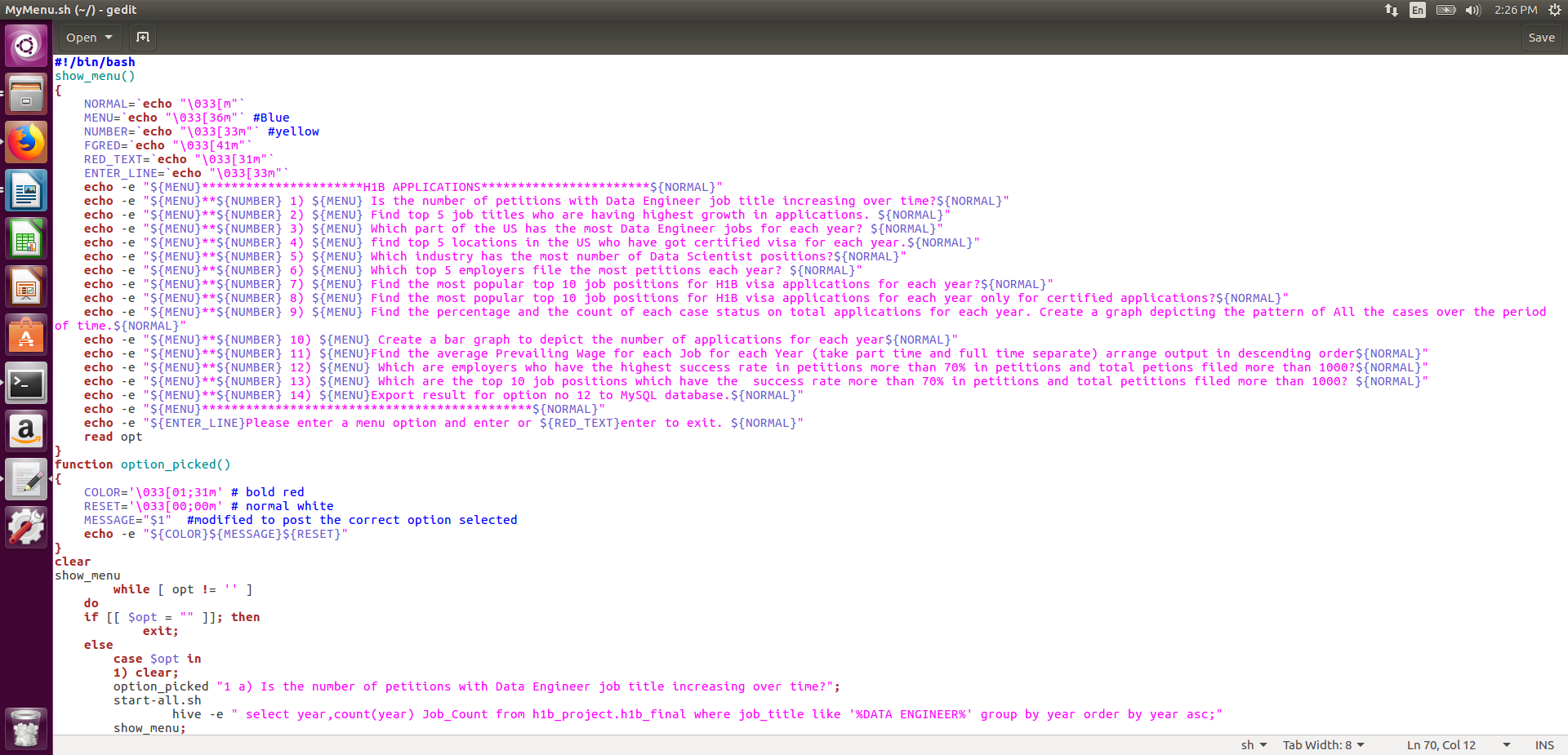
CREATE TABLE h1b\_final(s\_no int,case\_status string, employer\_name string, soc\_name string, job\_title string, full\_time\_position string,prevailing\_wage bigint,year string, worksite string, longitute double, latitute double )  
row format delimited  
fields terminated by '\t'  
STORED AS TEXTFILE;

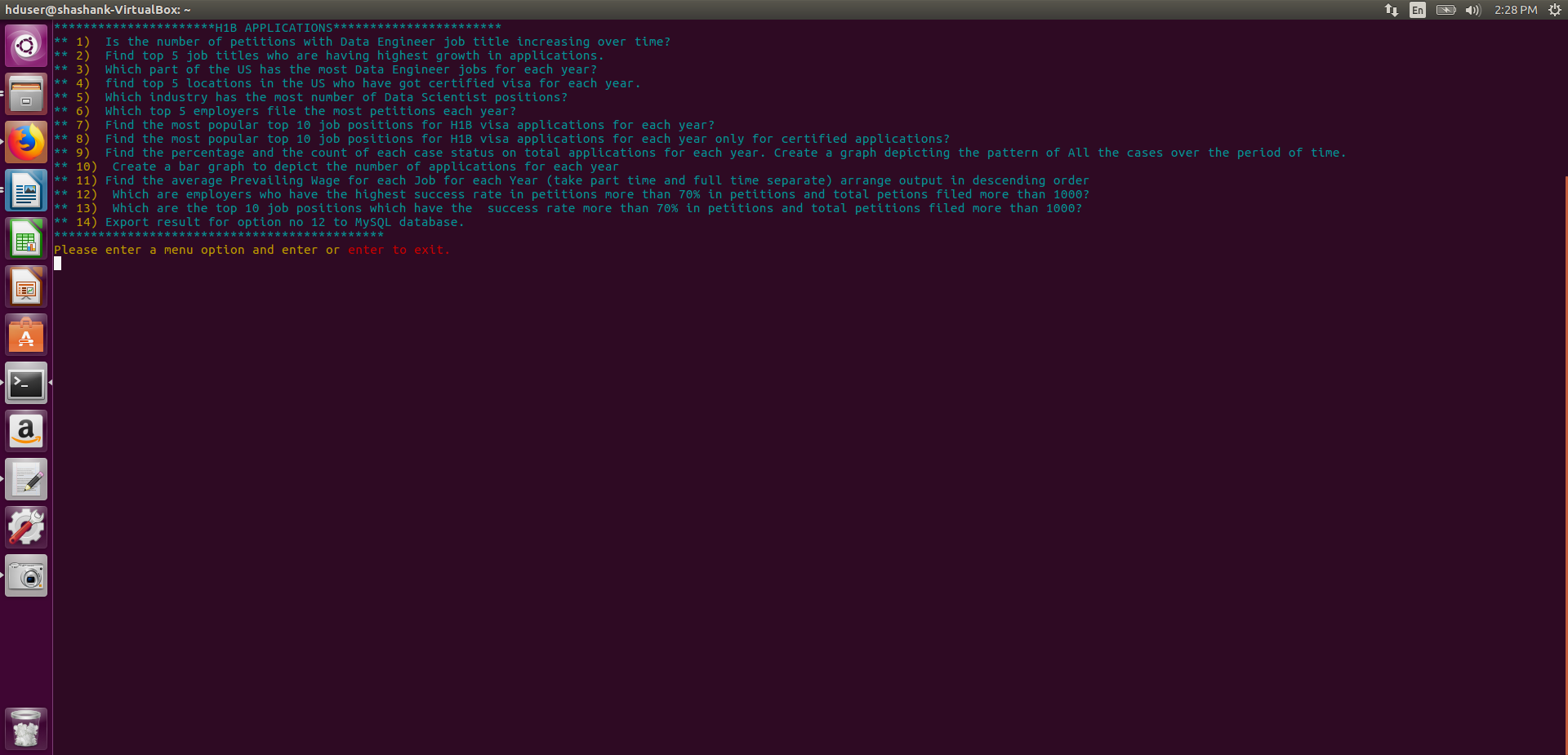
INSERT OVERWRITE TABLE h1b\_final SELECT s\_no,  
case when trim(case\_status) = "PENDING QUALITY AND COMPLIANCE REVIEW - UNASSIGNED" then "DENIED"  
when trim(case\_status) = "REJECTED" then "DENIED"  
when trim(case\_status) = "INVALIDATED" then "DENIED"  
else case\_status end,   
employer\_name, 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;

**Summary of work flow for application:**

* Data is imported into HDFS from flat file/csv for further cleaning and manipulation
* SQOOP interface is used to export data from PIG/HDFS to MYSQL file system
* Date is queried over for multiple user selection using hive & pig
* Linux shell scripting is used for console based UI
  + Code for shell scripting is available in the **MyMenu.sh** file.
    - **Steps to run the shell script:**
      * 1. Paste the shell file in the working directory of your system.
        2. Give permission like **chmod u+x MyMenu.sh**
        3. Run the script using **./MyMenu.sh**

**Snapshot of the Console based UI:**

****



**Implementation steps:**

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



Technology Used: Hive

Hive Script:

select year,count(year) Job\_Count from h1b\_project.h1b\_final where job\_title like '%DATA ENGINEER%' group by year order by year asc;

Sample Output:

Year job\_count

2011 60

2012 81

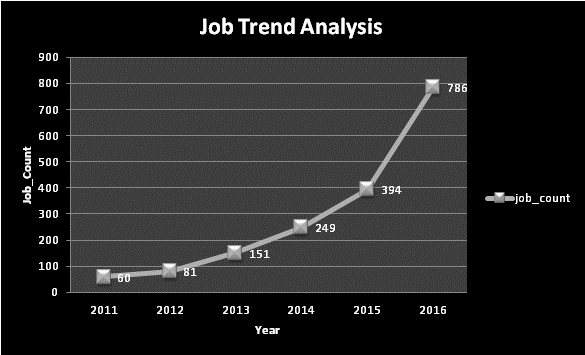
2013 151

2014 249

2015 394

2016 786

Given below graph shows Data Engineer job title increasing over time:



b) Find top 5 job titles who are having highest avg growth in applications.[ALL]

Technology Used: Pig

Pig Script:

data = LOAD '/home/hduser/DatasetsandCodes/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); --Load data

noheader= filter data by $0>=1; --Remove header

cleansed= filter noheader by $7 matches '2011'; --filtering dataset by year

a= group cleansed by $4; --grouping by job

step\_a= foreach a generate group,COUNT($1); --generate year,job,count

cleansed= filter noheader by $7 matches '2012'; --filtering dataset by year

a= group cleansed by $4; --grouping by job

step\_b= foreach a generate group,COUNT($1); --generate year,job,count

cleansed= filter noheader by $7 matches '2013'; --filtering dataset by year

a= group cleansed by $4; --grouping by job

step\_c= foreach a generate group,COUNT($1); --generate year,job,count

cleansed= filter noheader by $7 matches '2014'; --filtering dataset by year

a= group cleansed by $4; --grouping by job

step\_d= foreach a generate group,COUNT($1); --generate year,job,count

cleansed= filter noheader by $7 matches '2015'; --filtering dataset by year

a= group cleansed by $4; --grouping by job

step\_e= foreach a generate group,COUNT($1); --generate year,job,count

cleansed= filter noheader by $7 matches '2016'; --filtering dataset by year

a= group cleansed by $4; --grouping by job

step\_f= foreach a generate group,COUNT($1); --generate year,job,count

joined= join step\_a by $0,step\_b by $0,step\_c by $0,step\_d by $0,step\_e by $0,step\_f by $0;

yearwiseapplications= foreach joined generate $0,$1,$3,$5,$7,$9,$11;

--generate progressive growth

progressivegrowth= 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;

--average progressive growth

avgprogressivegrowth= foreach progressivegrowth generate $0,($1+$2+$3+$4+$5)/5;

---ordered progressive growth

orderedavggrowth= order avgprogressivegrowth by $1 desc;

--display top5 only

answer = limit orderedavggrowth 5;

dump answer;

Sample Output:

(SENIOR SYSTEMS ANALYST JC60,4255.4644)

(SOFTWARE DEVELOPER 2,3480.5925)

(PROJECT MANAGER 3,3233.3335)

(SYSTEMS ANALYST JC65,2984.8809)

(MODULE LEAD,2917.112)

2 a) which part of the US has the most Data Engineer jobs for each year?



Technology Used: Hive

Hive Script:

select split(worksite,'[,]')[1] as state,year,count(split(worksite,'[,]')[1]) as job\_cnt from h1b\_project.h1b\_final where job\_title LIKE '%DATA ENGINEER%' and year=$var group by year,split(worksite,'[,]')[1] order by year ;

Sample Output:

Enter the year (2011,2012,2013,2014,2015,**2016**)

state year job\_cnt

WASHINGTON 2016 140

UTAH 2016 5

TENNESSEE 2016 2

PENNSYLVANIA 2016 8

OREGON 2016 3

NORTH CAROLINA2016 4

NEW JERSEY 2016 14

MINNESOTA 2016 3

MICHIGAN 2016 7

MASSACHUSETTS 2016 33

MARYLAND 2016 14

IOWA 2016 4

INDIANA 2016 3

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



Technology Used: Hive

Hive Script:

select year, worksite,count(case\_status) as total\_case\_status from h1b\_project.h1b\_final where year =$var and case\_status='CERTIFIED' group by worksite,year order by total\_case\_status desc limit 5;

Sample Output:

Enter the year (2011,2012,2013,2014,2015,**2016**)

Year worksite total\_case\_status

2016 NEW YORK, NEW YORK 34639

2016 SAN FRANCISCO, CALIFORNIA13836

2016 HOUSTON, TEXAS 13655

2016 ATLANTA, GEORGIA 11678

2016 CHICAGO, ILLINOIS 11064

3)Which industry(SOC\_NAME) has the most number of Data Scientist positions? [certified]



Technology Used: Hive

Hive Script:

select soc\_name,count(soc\_name) as cnt from h1b\_project.h1b\_final where job\_title LIKE '%DATA SCIENTIST%' group by soc\_name order by cnt desc;

Sample Output:

soc\_name cnt

STATISTICIANS 649

COMPUTER AND INFORMATION RESEARCH SCIENTISTS 500

OPERATIONS RESEARCH ANALYSTS 426

Computer and Information Research Scientists 208

COMPUTER OCCUPATIONS, ALL OTHER 179

Statisticians 152

SOFTWARE DEVELOPERS, APPLICATIONS 148

MATHEMATICIANS 147

COMPUTER SYSTEMS ANALYSTS 135

4)Which top 5 employers file the most petitions each year? - Case Status - ALL



Technology Used: Hive

Hive Script:

select employer\_name, year,count(year) as cnt from h1b\_project.h1b\_final where year=$var group by year,employer\_name order by cnt desc limit 5;

Sample Output:

Enter the year (2011,2012,2013,2014,2015,**2016**)

employer\_name year cnt

INFOSYS LIMITED 2016 25352

CAPGEMINI AMERICA INC 2016 16725

TATA CONSULTANCY SERVICES LIMITED 2016 13134

WIPRO LIMITED 2016 10607

IBM INDIA PRIVATE LIMITED 2016 9787

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

a) for all the applications



Technology Used: Hive

Hive Script:

select job\_title,year,count(case\_status ) as temp from h1b\_project.h1b\_final where year= $var group by job\_title,year order by temp desc limit 10;

Sample Output:

Enter the year (2011,2012,2013,2014,2015,**2016**)

job\_title year temp

PROGRAMMER ANALYST 2016 53743

SOFTWARE ENGINEER 2016 30668

SOFTWARE DEVELOPER 2016 14041

SYSTEMS ANALYST 2016 12314

COMPUTER PROGRAMMER 2016 11668

BUSINESS ANALYST 2016 9167

COMPUTER SYSTEMS ANALYST 2016 6900

SENIOR SOFTWARE ENGINEER 2016 6439

DEVELOPER 2016 6084

TECHNOLOGY LEAD - US 2016 5410

b) for only certified applications.

Technology Used: Hive

Hive Script:

select job\_title,year,count(case\_status ) as temp from h1b\_project.h1b\_final where year= $var and case\_status='CERTIFIED' group by job\_title,year order by temp desc limit 10;

Sample Output:

Enter the year (2011,2012,2013,2014,2015,**2016**)

job\_title year temp

PROGRAMMER ANALYST 2016 47964

SOFTWARE ENGINEER 2016 25890

SOFTWARE DEVELOPER 2016 12474

SYSTEMS ANALYST 2016 10986

COMPUTER PROGRAMMER 2016 10528

BUSINESS ANALYST 2016 8175

COMPUTER SYSTEMS ANALYST 2016 6205

DEVELOPER 2016 5912

SENIOR SOFTWARE ENGINEER 2016 5630

TECHNOLOGY LEAD – US 2016 5405

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

Technology Used: Pig

Pig Script:

data = LOAD '/home/hduser/DatasetsandCodes/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); --load data

noheader= filter data by $0>=1; --remove header

cleansed= filter noheader by $1 is not null and $1!='NA';

temp= group cleansed by $7;

total= foreach temp generate group,COUNT(cleansed.$1); --describe total;

dump total;

noheader= filter data by $0>=1;

cleansed= filter noheader by $7 is not null and $7!='NA';

temp= group cleansed by ($7,$1);

yearsoccount= foreach temp generate group,group.$0,COUNT($1);

dump yearsoccount;

joined= join yearsoccount by $1,total by $0;

ans= foreach joined generate FLATTEN($0),(float)($2\*100)/$4,$2; --percent generation

dump ans;

Sample Output:

(2011,DENIED,8.119476,29130)

(2011,CERTIFIED-WITHDRAWN,3.2321813,11596)

(2011,WITHDRAWN,2.8165913,10105)

(2011,CERTIFIED,85.83175,307936)

(2012,WITHDRAWN,2.5805628,10725)

(2012,CERTIFIED-WITHDRAWN,7.487362,31118)

(2012,DENIED,5.075949,21096)

(2012,CERTIFIED,84.856125,352668)

(2013,WITHDRAWN,2.6214957,11590)

(2013,CERTIFIED-WITHDRAWN,8.014222,35432)

(2013,DENIED,2.7427316,12126)

(2013,CERTIFIED,86.61816,382951)

(2013,PENDING QUALITY AND COMPLIANCE REVIEW - UNASSIGNED,0.00339279,15)

(2014,DENIED,2.290216,11896)

(2014,CERTIFIED-WITHDRAWN,6.998096,36350)

(2014,INVALIDATED,1.9251983E-4,1)

(2014,WITHDRAWN,3.086863,16034)

(2014,CERTIFIED,87.624245,455144)

(2014,REJECTED,3.8503966E-4,2)

(2015,DENIED,1.765399,10923)

(2015,CERTIFIED,88.452255,547278)

(2015,WITHDRAWN,3.1443594,19455)

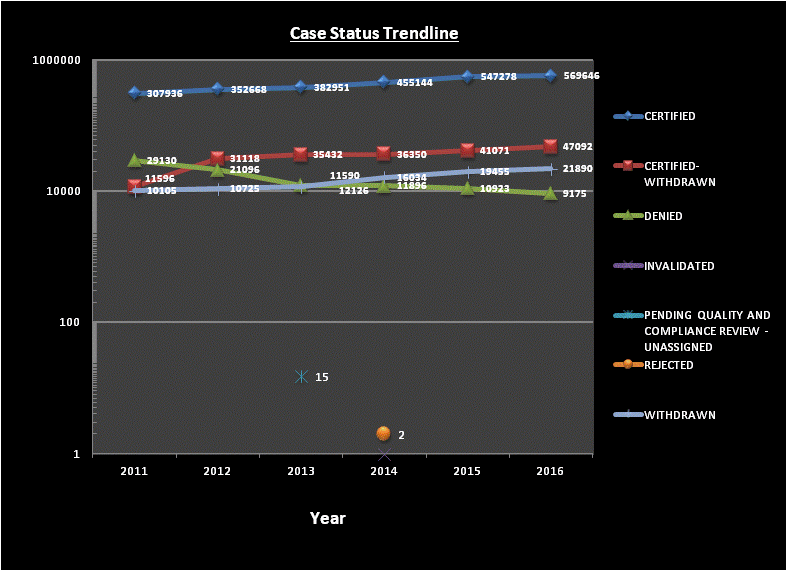
(2015,CERTIFIED-WITHDRAWN,6.6379843,41071)

(2016,DENIED,1.4163257,9175)

(2016,CERTIFIED,87.93507,569646)

(2016,WITHDRAWN,3.3791137,21890)

(2016,CERTIFIED-WITHDRAWN,7.269494,47092)



7) Create a bar graph to depict the number of applications for each year [All]



Technology Used: Hive

Hive Script:

select year,count(\*) as applications from h1b\_project.h1b\_final where year like '201%' group by year;

Sample Output:

year applications

2011 358767

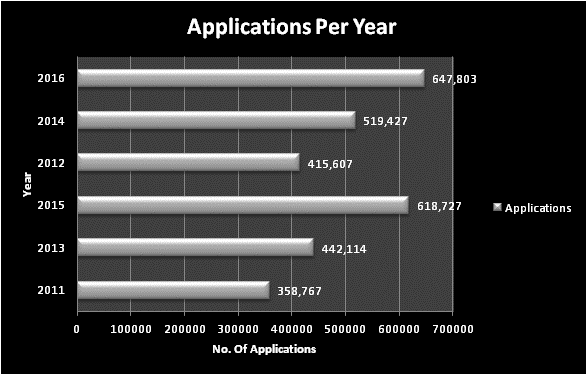
2013 442114

2015 618727

2012 415607

2014 519427

2016 647803



8) Find the average Prevailing Wage for each Job for each Year (take part time and full time separate). Arrange the output in descending order - [Certified and Certified Withdrawn.]



Technology Used: Hive

Hive Script:

select job\_title,full\_time\_position,case\_status,year,avg(prevailing\_wage) as average from h1b\_project.h1b\_final where full\_time\_position = '$var' and year = $year and (case\_status='CERTIFIED' or case\_status='CERTIFIED-WITHDRAWN') group by job\_title,full\_time\_position,case\_status,year order by average desc;

Sample Output:

Enter the year(2011,2012,2013,2014,2015,**2016**)

Enter the choice Full time/ Part time.(**Y**/N)

BIOSTATISTICIAN III Y CERTIFIED 2016 70013.0

BUSINESS OBJECTS ADMINISTRATOR/DEVELOPER Y CERTIFIED 2016 70013.0

DAM SOFTWARE DEVELOPER Y CERTIFIED 2016 70013.0

GLOBAL SUPPLY MANAGER, LOGISTICS MANAGEMENT Y CERTIFIED 2016 70013.0

DEMAND PLANNER, CUSTOMER SUPPLY CHAIN Y CERTIFIED 2016 70013.0

PURCHASING SPECIALIST, RETAIL DEVELOPMENT Y CERTIFIED 2016 70013.0

DOSIMETRIST Y CERTIFIED-WITHDRAWN 2016 70013.0

EBU SENIOR FINANCIAL ANALYST Y CERTIFIED 2016 70013.0

ENGINEER- EMBEDDED Y CERTIFIED 2016 70013.0

9) Which are the employers along with the number of petitions who have the success rate more than 70% in petitions. (total petitions filed 1000 OR more than 1000) ?



Technology Used: Pig

Pig Script:

data = LOAD '/home/hduser/DatasetsandCodes/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);

noheader= filter data by $0>=1; --Remove header

cleansed= filter noheader by $1 is not null and $1!='NA';

temp= group cleansed by $2;

total= foreach temp generate group,COUNT(cleansed.$1); --Group by employername and count the case statuses as a whole

certified= filter noheader by $1 == 'CERTIFIED';

temp1= group certified by $2;

totalcertified= foreach temp1 generate group,COUNT(certified.$1); --Group by employername and count the case status = 'CERTIFIED'

certified= filter noheader by $1 == 'CERTIFIED-WITHDRAWN';

temp2= group certified by $2;

totalcertifiedwithdrawn= foreach temp2 generate group,COUNT(certified.$1); --Group by employername and count the case status = 'CERTIFIED-WITHDRAWN'

joined= join totalcertified by $0,totalcertifiedwithdrawn by $0,total by $0;

--dump joined;

joined= foreach joined generate $0,$1,$3,$5;

intermediateoutput= foreach joined generate $0,(float)($1+$2)\*100/($3),$3;

intermediateoutput2= filter intermediateoutput by $1>70 and $2>1000; --Filter by success-rate greater than 70% and petition count above 1000

finaloutput= order intermediateoutput2 by $1 DESC;

dump finaloutput;

Sample Output:

(IBM INDIA PVT. LTD.,89.563866,1284)

(VIRTUSA CORPORATION,88.76861,2217)

(INDIANA UNIV. PURDUE UNIV. INDIANAPOLIS,88.57994,1007)

(THE UNIVERSITY OF ARIZONA,88.331726,1037)

(IBM CORPORATION,88.19675,13276)

(BROADCOM CORPORATION,88.12019,2862)

(IBM INDIA PRIVATE LIMITED,87.6852,34219)

(PERSISTENT SYSTEMS, INC.,87.06977,3225)

(AMERICAN INFORMATION TECHNOLOGY CORPORATION,86.67158,1358)

(CITRIX SYSTEMS, INC.,85.344826,1044)

(NETAPP, INC.,84.545456,1870)

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



Technology Used: Pig

Pig Script:

data = LOAD '/home/hduser/DatasetsandCodes/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);

noheader= filter data by $0>=1; --Remove header

--Count Total Applications

cleansed= filter noheader by $1 is not null and $1!='NA';

temp= group cleansed by $4;

total= foreach temp generate group,COUNT(cleansed.$1);

--Count Total Applications who are 'CERTIFIED'

certified= filter noheader by $1 == 'CERTIFIED';

temp1= group certified by $4;

totalcertified= foreach temp1 generate group,COUNT(certified.$1);

--Count Total Applications who are 'CERTIFIED-WITHDRAWN'

certified= filter noheader by $1 == 'CERTIFIED-WITHDRAWN';

temp2= group certified by $4;

totalcertifiedwithdrawn= foreach temp2 generate group,COUNT(certified.$1);

--SUCCESS\_RATE=(CERTIFIED+CERTIFIED-WITHDRAWN)/TOTAL X 100

joined= join totalcertified by $0,totalcertifiedwithdrawn by $0,total by $0;

joined= foreach joined generate $0,$1,$3,$5;

intermediateoutput= foreach joined generate $0,(float)($1+$2)\*100/($3),$3;

intermediateoutput2= filter intermediateoutput by $1>70 and $2>1000; --Filter by success-rate greater than 70% and petition count above 1000

finaloutput= order intermediateoutput2 by $1 DESC;

--STORE DATA INTO TEXT FILE

store finaloutput into '/home/hduser/Pig/question10' using PigStorage('\t');

Sample Output:

COMPUTER PROGRAMMER / CONFIGURER 2 100.0 1276

ASSOCIATE CONSULTANT - US 99.93171 4393

SYSTEMS ENGINEER - US 99.90036 10036

TEST ANALYST - US 99.818474 4958

CONSULTANT - US 99.81147 7426

TECHNOLOGY LEAD - US 99.80247 28350

TECHNICAL TEST LEAD - US 99.79531 5374

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

Technology Used: Sqoop

Sqoop Script:

hadoop fs -rm -r -f /Pig/Question10

hadoop fs -mkdir -p /Pig/Question10

hadoop fs -put /home/hduser/Pig/question10/p\* /Pig/Question10/

mysql -u root -p'shashank' -e 'drop database question11;create database if not exists question11;use question11;create table question11(job\_title varchar(100),success\_rate float,petitions int);';

sqoop export --connect jdbc:mysql://localhost/question11 --username root --password shashank --table question11 --update-mode allowinsert --export-dir /Pig/Question10/p\* --input-fields-terminated-by '\t' ;

echo -e '\n\nDisplay contents from MySQL Database.\n\n'

echo -e '\n10) Which are the top 10 job positions that have success rate more than 70% in petitions and total petitions filed more than 1000?\n\n'

mysql -u root -p'shashank' -e 'select \* from question11.question11';

Sample Output:

+------------------------------------------------------------+--------------+-----------+

| job\_title | success\_rate | petitions |

+------------------------------------------------------------+--------------+-----------+

| SOFTWARE DEVELOPERS, APPLICATIONS | 92.9707 | 1195 |

| COMPUTER SYSTEM ANALYST | 92.7525 | 3753 |

| OCCUPATIONAL THERAPIST | 92.7203 | 4437 |

| SENIOR APPLICATION DEVELOPER | 92.6209 | 1965 |

| SENIOR PROJECT MANAGER | 92.6108 | 1015 |

| CLINICAL FELLOW | 92.5829 | 1146 |

| INDUSTRIAL DESIGNER | 92.567 | 3619 |

**Conclusion**

Following is the conclusion that we can draw on the tasks performed by us:

1. The availability of Big Data, low-cost commodity hardware, and new information management and analytic software has the capabilities required to analyze huge data sets quickly and cost-effectively.
2. Sqoop is useful when to import and export data from SQL to Hadoop and vice versa
3. Hive helps in cleaning up of data and can be easily converted to csv format. Also all SQL queries can directly run on Hadoop using Hive.
4. For normal group by join, and filter based data retrieval, Pig is very efficient.

**Webography**

<https://drive.google.com/drive/folders/1KdTk0mQ-RJ7QzjgMw4w_tuBL2Bf0u8sy>

<http://www.student.niitcloudcampus.com/CloudCampus/Welcome.aspx>