**PROJECT REPORT**

**ON**

**“BIG DATA ANALYSIS USING HADOOP ON**

**H1B VISA DATA”**

**SUBMITED BY**

**RONAK CHUDASAMA**

**S180010900144**

**BATCH 162**

**TOWARDS FULLFILLMENT OF PROFESSIONAL DIPLOMA IN DIGITAL TRANSFORMATION - BIG DATA WITH HADOOP**

**SUBMITTED TO**

**NIIT Limited | Mumbai - Borivali**



**ACKNOWLEDGMENT**

I am using this opportunity to express my gratitude to everyone who supported me throughout the course big data certification project. I am thankful for their aspiring guidance, invaluably constructive criticism and friendly advice during the project work. I am sincerely grateful to them for sharing their truthful and illuminating views on a number of issues related to the project.

I express my warm thanks to Ms. Nandita Vora, Mr. Sandeep Agarwal and Mr. Annu Sharma for their support and guidance at NIIT Limited.

I would also like to thank Ms. Sonam Soni, Ms. Mitali and all the people who provided me facilities during project development.

Thank you,

Ronak Chudasama

**ABSTRACT**

The H-1B is a temporary (non-immigrant) visa category that allows employers to petition for highly educated foreign professionals to work in “specialty occupations” that require at least a bachelor’s degree or the equivalent. Jobs in fields such as mathematics, engineering, and technology often qualify. The U.S. employers submit their petitions on the first business day in April for the pool of H-1B visa numbers for which U.S. Citizenship and Immigration Services (USCIS) controls the allocation. The US immigration department receives over 200,000 petitions and selects 85,000 applications through a random process. The petitions data is available for public access to perform in-depth longitudinal research and analysis. This data provides key insights into the prevailing wages for job titles being sponsored by US employers under H1-B visa category. In our project analysis is carried out in order to get details about number of petitions for each job positions, top five job positions, top 5 locations, top ten employers and so on

|  |  |
| --- | --- |
| Topic | Page Number |
| Introduction To Big Data | 1 |
| Evolution of Big Data | 2 |
| Characteristics of Big Data | 3 |
| Need for Big Data | 4 |
| Introduction To Hadoop | 5 |
| Hadoop Ecosystem | 6-11 |
| Project Introduction | 12-13 |
| Dataset Description | 14-15 |
| Software and Hardware Requirement | 16 |
| Implementation Steps—Map Reduce | 17 |
| Query 1a | 19 |
| Query 1b | 20 |
| Query 2a | 21 |
| Query 2b | 25 |
| Query 3 | 28 |
| Query 4 | 31 |
| Query 5a | 33 |
| Query 5b | 36 |
| Query 6 | 36 |
| Query 7 | 39 |
| Query 8 | 39 |
| Query 9 | 40 |
| Query 10 | 41 |
| Query 11 | 42 |
| Conclusion | 43 |
| Webography | 44 |

**Introduction to Big Data**

According to the American IT research and advisory firm Gartner Inc. “Big Data is high-volume, high-velocity and/or high-variety information assets that demand cost-effective, innovative forms of information processing that enable enhanced insight, decision making, and process automation.”

Today, the size (volume), complexity (variety), and the rate of growth (velocity) of the data that organizations handle have reached such unbelievable levels that traditional processing and analytical tools fail quite miserably. Following are some of the sources of data explosion:

* A large stock exchange such as the New York Stock Exchange generates more than 1 terabyte of data daily.
* Worldwide, there are approximately 5 billion mobile phones in use; of these, nearly 1.75 billion are estimated to be smartphones.
* YouTube users upload more than 48 hours of video every minute. Every second of HD video generates bytes 2,000 times more than that required to store a single page of text.
* Large social networks like Twitter and Facebook generate more than 10 terabytes of data daily.
* There are more than 30 million networked sensors in the world, and each of them transmits data continuously.



Figure 1: Sources of Big Data

**Evolution of Big Data**

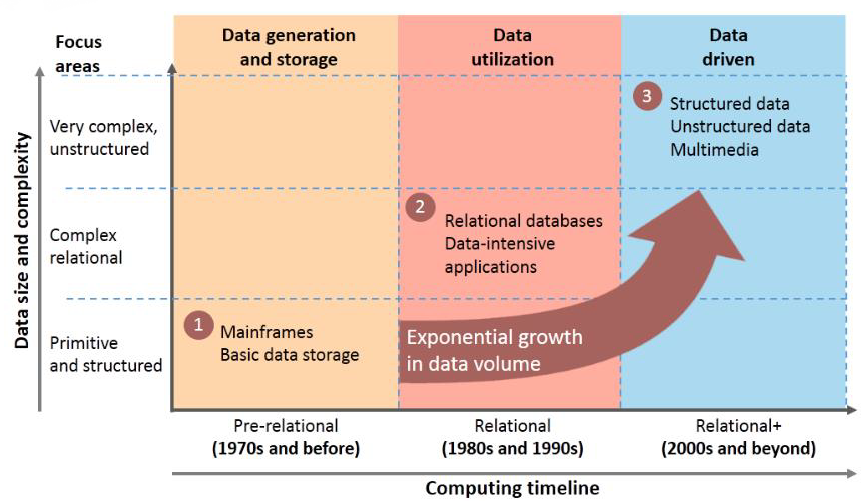


Figure 2: Evolution of Big Data

The Figure 2 gives a very real picture of the evolution of Big Data. Here, the data size and complexity have been mapped against computing timeline. In early 1970s, in fact even before that, the focus was on generation and storage of data. Mainframes were used for basic data storage. During the 1980s and 90s, the focus was on data utilization, for which relational databases and other data-intensive applications were brought into the picture. In the 2000s the spotlight was on data-driven technology, for which structured data, unstructured data, and multimedia were used. Throughout this evolution phase, we saw an exponential growth in data volume.

**Characteristics of Big Data**

The characteristics of Big Data are as follows:

* **Variety** encompasses managing the complexity of data in different structures, ranging from relational data to logs and raw text.
* **Velocity** accounts for managing the streaming of data and the movement of voluminous data at a high speed.
* **Volume** denotes managing the huge scaling of data ranging from terabytes to zettabytes.
* Voluminous data is useless without accuracy, as incorrect data causes concern in organizations. **Veracity** manages accuracy of data and its analyses especially in automated decision-making process. Data veracity is essential if the organization aims to be information-centric.
* Big Data is variable, which means the meaning of the data is constantly changing. This makes it relevant in sentiment analyses. To perform a proper sentiment analyses, algorithms need to understand the context and decipher the exact meaning of a word in that context.
* **Visualization** makes voluminous data comprehensible. With correct analyses and visualizations, raw data can used. Visualizations mean complex graphs that can include variables of data while still remaining understandable and readable.
* The value of data is in analyses, which turns data into information, followed by knowledge. Based on data value, organizations can become information-centric.

**Need for Big Data**

According to an estimation approximately 90% of the world’s data has been created in the last two years. It includes 10% of structured data and 90% of unstructured data that are difficult to analyze. The structured formats, such as databases, have limitations while handling large data sets and there is difficulty in integrating distributed information. Furthermore, most business users are not aware of the requirements during IT system development.

Potentially valuable data for varied systems like Enterprise Resource Planning or ERP and Supply Chain Management or SCM, are either dormant or discarded forming a data puddle within enterprises. It is often too expensive to justify the integration of large volumes of unstructured data.

Therefore, Big Data is needed to analyze and integrate the enterprise’s large data sets irrespective of the data types.

**Introduction to Hadoop**

The Apache Hadoop project develops open-source software for reliable, scalable, distributed computing.

The Apache Hadoop software library is a framework that allows for the distributed processing of large data sets across clusters of computers using simple programming models. It is designed to scale up from single servers to thousands of machines, each offering local computation and storage. Rather than rely on hardware to deliver high-availability, the library itself is designed to detect and handle failures at the application layer, so delivering a highly-available service on top of a cluster of computers, each of which may be prone to failures.

The project includes these modules:

* **Hadoop Common**: The common utilities that support the other Hadoop modules.
* **Hadoop Distributed File System (HDFS™)**: A distributed file system that provides high-throughput access to application data.
* **Hadoop YARN**: A framework for job scheduling and cluster resource management.
* **Hadoop MapReduce**: A YARN-based system for parallel processing of large data sets.

**Hadoop Ecosystem**

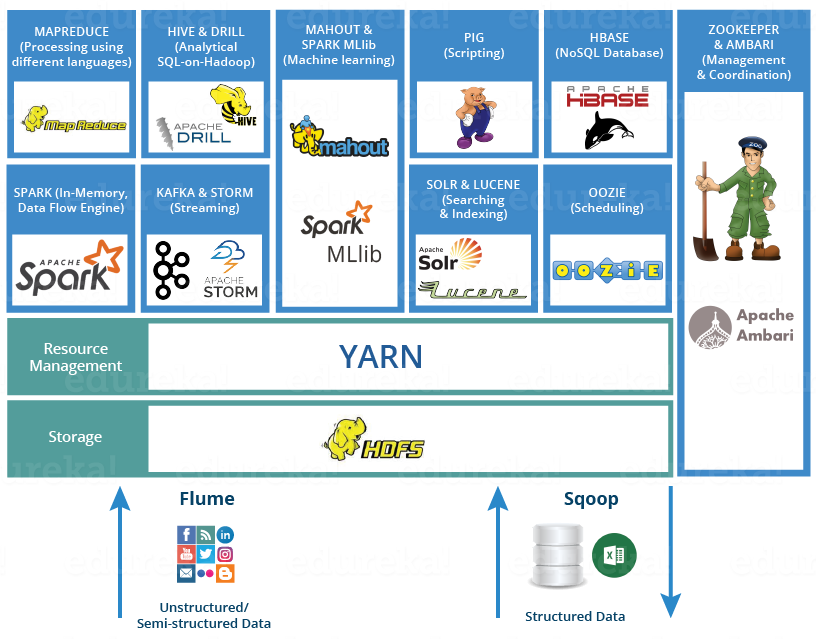


Figure 2 Hadoop Ecosystem

## ****HDFS****

* [**Hadoop Distributed File System**](https://www.edureka.co/blog/hdfs-tutorial) is the core component or we can say, the backbone of Hadoop Ecosystem.
* It helps us in storing our data across various nodes and maintaining the log file about the stored data (metadata).
* HDFS has two core components, i.e. **NameNode and DataNode**.
  1. The **NameNode** is the main node and it doesn’t store the actual data. It contains metadata, just like a log file or you can say as a table of content. Therefore, it requires less storage and high computational resources.
  2. On the other hand, all your data is stored on the **DataNodes** and hence it requires more storage resources. These DataNodes are commodity hardware (like your laptops and desktops) in the distributed environment. That’s the reason, why Hadoop solutions are very cost effective.
  3. Client always communicate to the NameNode while writing the data. Then, it internally sends a request to the client to store and replicate data on various DataNodes.

**YARN**

Consider YARN as the brain of your Hadoop Ecosystem. It performs all your processing activities by allocating resources and scheduling tasks. It has two major components, i.e. ResourceManager and NodeManager.

**MAPREDUCE**

It is the core component of processing in a Hadoop Ecosystem as it provides the logic of processing. In other words, MapReduce is a software framework which helps in writing applications that processes large data sets using distributed and parallel algorithms inside Hadoop environment.

## ****APACHE PIG****

* PIG has two parts: **Pig Latin**, the language and **the pig runtime,** for the execution environment. We can better understand it as Java and JVM.
* It supports*pig latin* language, which has SQL like command structure.

## ****APACHE HIVE****

* Facebook created HIVE for people who are fluent with SQL. Thus, HIVE makes them feel at home while working in a Hadoop Ecosystem.
* Basically, HIVE is a data warehousing component which performs reading, writing and managing large data sets in a distributed environment using SQL-like interface.
* ***HIVE + SQL = HQL***
* The query language of Hive is called Hive Query Language(HQL), which is very similar like SQL.
* It has 2 basic components: **Hive Command Line and JDBC/ODBC driver**.

**APACHE MAHOUT**

* Mahout provides an environment for creating machine learning applications which are scalable.
* It performs **collaborative filtering, clustering and classification.**Some people also consider**frequent item set missing**as Mahout’s function.

## ****APACHE SPARK****

* Apache Spark is a framework for real time data analytics in a distributed computing environment.
* The Spark is written in Scala and was originally developed at the University of California, Berkeley.
* It executes in-memory computations to increase speed of data processing over Map-Reduce.
* It is 100x faster than Hadoop for large scale data processing by exploiting in-memory computations and other optimizations. Therefore, it requires high processing power than Map-Reduce.

## ****APACHE HBASE****

* HBase is an open source, non-relational distributed database. In other words, it is a NoSQL database.
* It supports all types of data and that is why, it’s capable of handling anything and everything inside a Hadoop ecosystem.
* It gives us a fault tolerant way of storing sparse data, which is common in most Big Data use cases.
* The HBase is written in Java, whereas HBase applications can be written in REST, Avro and Thrift APIs.

**APACHE DRILL**

* Apache Drill is used to drill into any kind of data. It’s an open source application which works with distributed environment to analyze large data sets.
* Basically, the main aim behind Apache Drill is to provide scalability so that we can process petabytes and exabytes of data efficiently (or you can say in minutes).
* The main power of Apache Drill lies in ***combining a variety of data stores just by using a single query.***

## ****APACHE ZOOKEEPER****

* Apache Zookeeper is the coordinator of any Hadoop job which includes a combination of various services in a Hadoop Ecosystem.
* Apache Zookeeper coordinates with various services in a distributed environment

## ****APACHE OOZIE****

Consider Apache Oozie as a clock and alarm service inside Hadoop Ecosystem. For Apache jobs, Oozie has been just like a scheduler. It schedules Hadoop jobs and binds them together as one logical work.

## ****APACHE FLUME****

The Flume is a service which helps in ingesting unstructured and semi-structured data into HDFS.

**APACHE SQOOP**

 The major difference between Flume and Sqoop is that:

* Flume only ingests unstructured data or semi-structured data into HDFS.
* While Sqoop can import as well as export structured data from RDBMS or Enterprise data warehouses to HDFS or vice versa.

## ****APACHE SOLR & LUCENE****

Apache Solr and Apache Lucene are the two services which are used for searching and indexing in Hadoop Ecosystem.

## ****APACHE AMBARI****

* Ambari is an Apache Software Foundation Project which aims at making Hadoop ecosystem more manageable.
* It includes software for **provisioning, managing and monitoring** Apache Hadoop clusters.

**Project 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.

We will be performing analysis on the H1B visa applicants between the years 2011-2016. The dataset has nearly 3 million records. After analysing the data, we can derive the following facts.

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 average growth in applications.

[Case Status-ALL]

2 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.[Case Status -Certified]

3)Which industry(SOC\_NAME) has the most number of Data Scientist positions?[Case Status -Certified]

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

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

a) for all the applications

b) for only certified applications.

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.

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

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.]

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) ?

SUCCESS RATE % = (Certified + Certified Withdrawn)/Total x 100

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)?

SUCCESS RATE % = (Certified + Certified Withdrawn)/Total x 100

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

**Dataset Description**

|  |  |  |
| --- | --- | --- |
| Sr No. | Column Name | Description |
| 1 | CASE\_STATUS | Status associated with the last significant event or decision. Valid values include “Certified,” “Certified-Withdrawn,” Denied,” and “Withdrawn”.  Certified: Employer filed the LCA(Labour Condition Application), which was approved by DOL(Department of Labor)  Certified Withdrawn: LCA was approved but later withdrawn by employer Withdrawn: LCA was withdrawn by employer before approval  Denied: LCA was denied by DOL |
| 2 | EMPLOYER\_NAME | Name of employer submitting Labour 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 labour condition, as classified by the Standard Occupational Classification (SOC) System. |
| 4 | JOB\_TITLE | Title of the job |
| 5 | FULL\_TIME\_POSITION | Y = Full Time Position; N = Part Time Position |
| 6 | 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. |
| 7 | YEAR | Year in which the H1B visa petition was filed |
| 8 | WORKSITE | City and State information of the foreign worker’s intended area of employment |
| 9 | LONGITUDE | Longitude of the Worksite. |
| 10 | LATITUDE | Latitude of the Worksite. |

**Software Requirement**

* Ubuntu16.04 LTS
* Hadoop 2.6.0
* Hive 1.2.1
* Apache Pig version 0.13.0
* mysql Ver 14.14 Distrib 5.7.20, for Linux (x86\_64)
* Sqoop 1.4.6
* java-8-openjdk-amd64
* Eclipse 3.8.1

**Hardware Requirement**

* Minimum 2GB RAM
* 1TB Hard Drive
* intel i5 processor

**Implementation Steps**

**MapReduce Programming Steps**

1. Import required Hadoop Libraries.
2. Define a class which extends Mapper class and provide Input and Output type for Key-Value pair.
3. Read records line by line inside map( ) method of Mapper class. Usually splitting the records line and extracting only required field.
4. Pass required field as Key-Value pair for further processing.
5. Define Partitioner class to segregate the output of Mapper class to different reduce function( Only if required).
6. Define a Reducer class which extends Reducer class and provide Output key value pair type for final result.
7. Process records received from Mapper or Partitioner class in reduce( ) method of Reducer class. Usually aggregation logic is coded here.

In this project query implemented using MapReduce programming:

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

over time?

2 a) Which part of the US has the most Data Engineer jobs for each

year?

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

3) Which industry (SOC\_NAME) has the most number of Data

Scientist positions? [Case Status: Certified]

**Hive Queries**

Hive provides us data **warehousing** facilities on top of an existing Hadoop cluster. Along with that it provides an **SQL like** interface which makes your work easier, in case you are coming from an SQL background. You can create tables in Hive and store data there.

In this project query implemented using Hive:

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

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

5 b) Find the most popular top 10 job positions for H1B visa applications for each year for only certified applications.

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

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

a. Mapper class reads data line by line in map() method to form key-value pair of records. The only required records is one which contains job title as Data Engineer. Key-Value pair of records is [ Year-Count].

b. The output of map() method is internally passed to sorting and shuffling which sorts key-value in ascending order of key and forms a group based on key. Here grouping will based on key as year.

c. Reducer class receives records from sort & shuffle phase and performs aggregation for each record in reduce() method i.e calculates total petition for each year.

d. After calculation of total petitions for each year inside cleanup() method of reducer class average growth is calculated by using formula as:

**( current year petitions - previous year petitions ) / previous year**

**petitions)\*100**

Mapper Phase(Key-Year, Value-Count For Data Engineer Job )

2012 1

2011 1

2012 1

2011 1

2012 1

2013 1

2013 1

Sort phase(Key-Value Sorted By Key)

2011 1

2011 1

2012 1

2012 1

2012 1

2013 1

2013 1

Shuffle phase(Key-Value Shuffled By Key)

2011 {1, 1}

2012 { 1, 1, 1}

2013 {1,1}

Reduce Phase

reduce() method(Aggregation For Each Key)

2011 2

2012 3

2013 2

cleanup() method(Growth Calculation)

1. 50 -33.33

**Expected Result:**

year0 0.0

year1 35.0

year2 86.41975308641975

year3 64.90066225165563

year4 58.23293172690763

year5 99.49238578680203

**1 b) Find top 5 job titles who are having highest average growth in applications.**

1. Load the data from Part1 and Part2.csv file.
2. Filter bag data2011 by year= 2011.
3. Group filter data on job\_title.
4. Count applications for that job\_title and year.
5. Repeat the steps for years 2012, 2013, 2014, 2015, 2016
6. Join the data for all years 2011, 2012, 2013, 2014, 2015, 2016 using common joining key.
7. Calculate average growth from joined data
8. Arrange the result in descending order to get top 5 job\_title.

**Expected Result:**

senior systems analyst jc60,4255.4

software developer 2,3480.8

project manager 3,3233.4

systems analyst jc65,2985.0

module lead,2917.2

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

a. Mapper class reads data line by line in map( ) method to form key-value pair of records. The only required records is one which contains job title as Data Engineer and case status as CERTIFIED. Key-Value pair of records is [Worksite-Count].

e. Partitioner class receives each records and divides the records for reduce phase based on year by getPartition( ) method. Each partition will have particular year data only.

c. The output of getPartition( ) method is passed to sorting and shuffling which sorts key-value in ascending order of key and forms a group based on key. Here grouping will based on key as worksite. This is how actually individual years data gets generated and sorted.

c. Reducer class performs aggregation for each record in reduce( ) method i.e calculates total application for each worksite in each partition.

d. After calculation of total applications for each worksite, only 5 worksites which has highest total applications are sent as output of reduce( ) method. This sorting of worksites having highest number of applications are filtered by java collection class TreeMap.

Mapper Phase(Key-Worksite, Value-Count of Data Enginneer Job)

ABC 1

PQR 1

ABC 1

XYZ 1

ABC 1

PQR 1

PQR 1

Partitioner Phase

Key Value pairs portioned based on field Year. Separate partition for each year. Each partition will receive that specific key-value pair. This partition key value pairs will be passed to sorting, shuffling and reduce phase individually.

Sort phase(Key-Value Sorted By Key-Worksite)

ABC 1

ABC 1

ABC 1

PQR 1

PQR 1

PQR 1

XYZ 1

Shuffle phase(Key-Value Shuffled By Key-Worksite)

ABC { 1, 1, 1}

PQR { 1, 1, 1}

XYZ {1}

Reduce Phase

reduce() method(Aggregation For Each Key-Worksite)

ABC 3

PQR 3

XYZ 1

Each partition will have worksite name with its total petition for job post data engineer for specific year.

**Expected Result**

seattle, washington 2011,19

new york, new york 2011,4

san francisco, california 2011,3

waltham, massachusetts 2011,2

tallahassee, florida 2011,1

seattle, washington 2012,26

san francisco, california 2012,6

pontiac, michigan 2012,3

san mateo, california 2012,2

woodland hills, california 2012,1

seattle, washington 2014,42

san francisco, california 2014,28

menlo park, california 2014,18

new york, new york 2014,16

mountain view, california 2014,13

seattle, washington 2015,60

san francisco, california 2015,52

new york, new york 2015,35

menlo park, california 2015,21

mountain view, california 2015,17

**2 b) find top 5 locations in the US who have got certified visa for each year. [Certified]**

a. Mapper class reads data line by line in map( ) method to form key-value pair of records. The only required records is one which contains case status as CERTIFIED. Key-Value pair of records is [Worksite-Count].

e. Partitioner class receives each records and divides the records for reduce phase based on year by getPartition( ) method. Each partition will have particular year data only.

c. The output of getPartition( ) method is passed to sorting and shuffling which sorts key-value in ascending order of key and forms a group based on key. Here grouping will based on key as worksite. This is how actually individual years data gets generated and sorted.

c. Reducer class performs aggregation for each record in reduce( ) method i.e calculates total application for each worksite in each partition.

d. After calculation of total applications for each worksite, only 5 worksites which has highest total applications are sent as output of reduce( ) method. This sorting of worksites having highest number of applications are filtered by java collection class TreeMap. This filteration of worksite having highest number of applications are done by java collection class TreeMap.

Mapper Phase(Key-Worksite, Value-Count of Certfied Status Jobs)

ABC 1

PQR 1

ABC 1

XYZ 1

ABC 1

PQR 1

PQR 1

Partitioner Phase

Key Value pairs portioned based on field Year. Separate partition for each year. Each partition will receive that specific key-value pair. This partition key value pairs will be passed to sorting, shuffling and reduce phase individually.

Sort phase(Key-Value Sorted By Key-Worksite)

ABC 1

ABC 1

ABC 1

PQR 1

PQR 1

PQR 1

XYZ 1

Shuffle phase(Key-Value Shuffled By Key-Worksite)

ABC { 1, 1, 1}

PQR { 1, 1, 1}

XYZ {1}

Reduce Phase

reduce() method(Aggregation For Each Key-Worksite)

ABC 3

PQR 3

XYZ 1

Each partition will have worksite name with its total petition of certified application for specific year.

**Expected Result:**

new york, new york 2011,23172

houston, texas 2011,8184

chicago, illinois 2011,5188

san jose, california 2011,4713

san francisco, california 2011,4711

new york, new york 2012,23737

houston, texas 2012,9963

san francisco, california 2012,6116

chicago, illinois 2012,5671

atlanta, georgia 2012,5565

new york, new york 2013,23537

houston, texas 2013,11136

san francisco, california 2013,7281

san jose, california 2013,6722

atlanta, georgia 2013,6377

new york, new york 2014,27634

houston, texas 2014,13360

san francisco, california 2014,9798

san jose, california 2014,8223

atlanta, georgia 2014,8213

new york, new york 2015,31266

houston, texas 2015,15242

san francisco, california 2015,12594

atlanta, georgia 2015,10500

san jose, california 2015,9589

new york, new york 2016,34639

san francisco, california 2016,13836

houston, texas 2016,13655

atlanta, georgia 2016,11678

chicago, illinois 2016,11064

**3) Which industry (SOC\_NAME) has the most number of Data Scientist positions?[Case Status: Certified]**

Technology used: MapReduce job in Java

Implementation Steps

a. Mapper class reads data line by line in map( ) method to form key-value pair of records. The only required soc\_name which contains job title as Data Engineer and case status as CERTIFIED. Key-Value pair of records is [soc\_name-Count].

e. Partitioner class receives each records and divides the records for reduce phase based on year by getPartition( ) method. Each partition will have particular year data only.

c. The output of getPartition( ) method is passed to sorting and shuffling which sorts key-value in ascending order of key and forms a group based on key. Here grouping will based on key as soc\_name. This is how actually individual years data gets generated and sorted.

c. Reducer class performs aggregation for each record in reduce( ) method i.e calculates total application for each soc name having data scientist job in each partition.

d. After calculation of total applications for each soc name, only 5 soc name which has highest total applications are sent as output of reduce( ) method. This filteration of soc name having highest number of data scientist are done by java collection class TreeMap.

Mapper Phase(Key-SOC\_NAME, Value-Count of Certfied Status

Data Scientist Job)

ABC 1

PQR 1

ABC 1

XYZ 1

ABC 1

PQR 1

PQR 1

Partitioner Phase

Key Value pairs portioned based on field Year. Separate partition for each year. Each partition will receive that specific key-value pair. This partition key value pairs will be passed to sorting, shuffling and reduce phase individually.

Sort phase(Key-Value Sorted By Key- SOC\_NAME)

ABC 1

ABC 1

ABC 1

PQR 1

PQR 1

PQR 1

XYZ 1

Shuffle phase(Key-Value Shuffled By Key- SOC\_NAME)

ABC { 1, 1, 1}

PQR { 1, 1, 1}

XYZ {1}

Reduce Phase

reduce() method(Aggregation For Each Key- SOC\_NAME)

ABC 3

PQR 3

XYZ 1

Each partition will have SOC\_NAME with its total petition of certified application of data scientist job for specific year.

**Expected Result:**

STATISTICIANSDATA SCIENTIST,705

COMPUTER AND INFORMATION RESEARCH SCIENTISTSDATA SCIENTIST,600

OPERATIONS RESEARCH ANALYSTSDATA SCIENTIST,484

COMPUTER OCCUPATIONS, ALL OTHERDATA SCIENTIST,203

SOFTWARE DEVELOPERS, APPLICATIONSDATA SCIENTIST,174

MATHEMATICIANSDATA SCIENTIST,154

COMPUTER SYSTEMS ANALYSTSDATA SCIENTIST,148

SOFTWARE DEVELOPERS, SYSTEMS SOFTWAREDATA SCIENTIST,69

BIOLOGICAL SCIENTISTS, ALL OTHERDATA SCIENTIST,31

DATABASE ADMINISTRATORSDATA SCIENTIST,22

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

a. In hive create table top5\_employer\_2011 which has columns as year, employer\_name, number\_of\_petitions.

b. Insert records from partitioned table data\_partitioned for required year by counting total petitions filed of each employers.

c. Store five employers details in table which has filed maximum number of petitions.

d. Similarly Create table top5\_employer\_2012, top5\_employer\_2013, top5\_employer\_2014, top5\_employer\_2015, top5\_employer\_2016.

e. Repeat steps b and c for tables created in step d.

**Expected Result**

top5\_employer\_2011.year top5\_employer\_2011.employer\_name top5\_employer\_2011.number\_of\_petitions

2011 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

top5\_employer\_2012.year top5\_employer\_2012.employer\_name top5\_employer\_2012.number\_of\_petitions

2012 infosys limited 15818

2012 wipro limited 7182

2012 tata consultancy services limited 6735

2012 deloitte consulting llp 4727

2012 ibm india private limited 4074

top5\_employer\_2013.year top5\_employer\_2013.employer\_name top5\_employer\_2013.number\_of\_petitions

2013 infosys limited 32223

2013 tata consultancy services limited 8790

2013 wipro limited 6734

2013 deloitte consulting llp 6124

2013 accenture llp 4994

top5\_employer\_2014.year top5\_employer\_2014.employer\_name top5\_employer\_2014.number\_of\_petitions

2014 infosys limited 23759

2014 tata consultancy services limited 14098

2014 wipro limited 8365

2014 deloitte consulting llp 7017

2014 accenture llp 5498

top5\_employer\_2015.year top5\_employer\_2015.employer\_name top5\_employer\_2015.number\_of\_petitions

2015 infosys limited 33245

2015 tata consultancy services limited 16553

2015 wipro limited 12201

2015 ibm india private limited 10693

2015 accenture llp 9605

top5\_employer\_2016.year top5\_employer\_2016.employer\_name top5\_employer\_2016.number\_of\_petitions

2016 infosys limited 25352

2016 capgemini america inc 16725

2016 tata consultancy services limited 13134

2016 wipro limited 10607

2016 ibm india private limited 9787

**5 a) Find the most popular top 10 job positions for H1B visa applications for each year for all the applications.**

a. In hive create table top10\_job\_position\_2011 which has columns as year, job\_position, number\_of\_applications.

b. Insert records from partitioned table data\_partitioned for required year by counting total applications per job\_position.

c. Store ten job position details in table which has maximum number of applications.

d. Similarly Create table top10\_job\_position\_2012, top10\_job\_position\_2013, top10\_job\_position\_2014, top10\_job\_position\_2015, top10\_job\_position\_2016.

e. Repeat steps b and c for tables created in step d.

**Expected Result**

result.year result.job\_position result.number\_of\_applications

2011 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

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

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

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

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

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

**5 b) Find the most popular top 10 job positions for H1B visa applications for each year for only certified applications.**

1. In hive create table top10\_job\_position\_2011\_certified which has columns as year, job\_position, number\_of\_applications.

b. Insert records from partitioned table data\_partitioned for required year by filtering case\_status as ‘CERTIFIED’ and counting total applications per job\_position .

c. Store ten job position details in table which has maximum number of applications.

d. Similarly Create table top10\_job\_position\_2012, top10\_job\_position\_2013, top10\_job\_position\_2014, top10\_job\_position\_2015, top10\_job\_position\_2016.

e. Repeat steps b and c for tables created in step d.

**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.**

* 1. Load the data into bag1 from Part1 and Part2.csv file and filter data on year!=’NA’.
  2. Filter data of bag requiredFileds by case\_status=’CERTIFIED’ and insert into bag onlyCertified.
  3. Group data of bag onlyCertified by year.
  4. Count total applications on Year for bag onlyCertified.
  5. Filter data of bag requiredFileds by case\_status=’CERTIFIED-WITHDRAWN’ and insert into bag onlyCertifiedWithdrawn.
  6. Group data of bag onlyCertifiedWithdrawn by year.
  7. Count total applications on Year for bag onlyCertifiedwithdrawn.
  8. Filter data of bag requiredFileds by case\_status=’DENIED’ and insert into bag onlyDenied.
  9. Group data of bag onlyDenied by year.
  10. Count total applications on Year for bag onlyDenied.
  11. Filter data of bag requiredFileds by case\_status=’Withdrawn’ and insert into bag onlyWithdrawn.
  12. Group data of bag onlyWithdrawn by year.
  13. Count total applications on Year for bag onlyWithdrawn.
  14. Join the data using common joining key.
  15. Calculate percentage from joined data.
  16. Create line graph on result.

**Expected Result**

Year, Certfied, Certified Withdrawn, Withdrawn, Denied, Total, Certified %, Certified Withdrawn %, Withdrawn %, Denied %

2011,307936,11596,10105,29130,358767,85.83,3.23,2.82,8.12

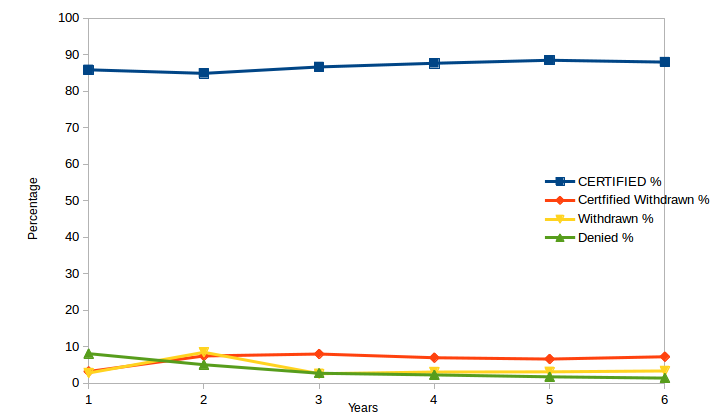
2012,352668,31118,10725,21096,415607,84.86,7.49,2.58,5.08

2013,382951,35432,11590,12141,442114,86.62,8.01,2.62,2.75

2014,455144,36350,16034,11899,519427,87.62,7.0,3.09,2.29

2015,547278,41071,19455,10923,618727,88.45,6.64,3.14,1.77

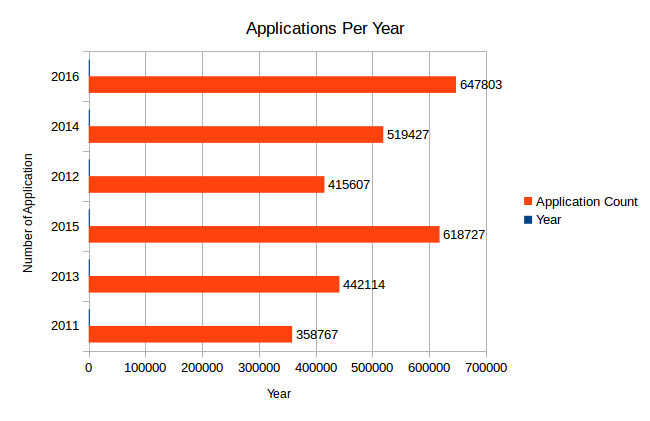
2016,569646,47092,21890,9175,647803,87.94,7.27,3.38,1.42



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

* + - * 1. Select year, count(\*) from h1b\_final
        2. Group by year
        3. Generate bar graph

**Expected Graph**



**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.]**

a. Load data into bag

b. Filter bag by year 2011, full time position Y and case status as Ceritified or certified-withdrawn.

c. Group filtered data by year and job title

d. Calculate avaerage of prevailing wage for each group.

e. Repeat step b to d for year 2012, 2013, 2014, 2015 and 2016.

f. Filter bag by year 2011, full time position N and case status as Ceritified or certified-withdrawn.

g. Group filtered data by year and job title

h. Calculate avaerage of prevailing wage for each group.

**Expected Result:**

**Full Time**

2016,TRAUMA & GENERAL SURGEON,328972.0

2016,SYSTEMS ANALYSTS,4216025.545454546

2016, MEDICAL ONCOLOGIST AND MEDICAL DIRECTOR, 292138.0

2016,CARDIOLOGIST/INTERVENTIONALCARDIOLOGIST, 350000.0

**PartTime**

2011,AUSTRALIAN OFFSHORE DRILLING REGULATION SPECIALIST,416000.0

2011,INTERNATIONAL MERCHANDISE SALES MANAGER,4197440.0

2011,LECTURER IN ARCHITECTURE,405600.0

2011,OCCUPATIONAL SAFETY AND HEALTH EXPERT,416000.0

**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) ?**

a. Load data into bag.

b.Filter bag by case status certified.

c. Group filtered data by employer name

d. Count total application for each group i.e employer name

e. Repeat Step c and d after filtering bag for case\_status certified withdrawn.

f. Calculate total application for each employer name[ This will consider all case status]

h. Join bags having total applications for each case status and total application(Considering all case status) by common filed i.e employer name.

i. Calculate success rate by formula certified (certified count + certified withdrawn count)/Total application)\*100 for each employer name.

k. Dump employer name, certified count, certified withdrawn count,total application, success rate for records which has success rate >70 and total applications >=1000.

**Expected Result:**

Employer name, Success Rate, Total Count

hcl america, inc.,99,22678

reliable software resources, inc.,99,1992

tata consultancy services limited,99,64726

erp analysts, inc.,99,1785

**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)?**

a. Load data into bag.

b. Filter bag by case status certified.

c. Group filtered data by job positions

d. Count total application for each group i.e job positions

e. Repeat Step c and d after filtering bag for case\_status certified withdrawn.

f. Calculate total application for each job positions[ This will consider all case status]

h. Join bags having total applications for each case status and total application(Considering all case status) by common filed i.e job positions

i. Calculate success rate by formula certified (certified count + certified withdrawn count)/Total application)\*100 for each job positions

k. Dump job positions, certified count, certified withdrawn count,total application, success rate for records which has success rate >70 and total applications >=1000.

**Expected Result**

Job Position, Success Rate, Total Count

COMPUTER PROGRAMMER / CONFIGURER 2,100,1276

PROJECT MANAGER - US,99,7046

TECHNOLOGY ANALYST - US,99,26055

SYSTEMS ANALYST - II,99,1339

PROGRAMMER ANALYST - II,99,3588

TECHNOLOGY LEAD - US,99,28350

PROJECT MANAGER - III,99,1651

**Sqoop Implementation**

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

1. Create database h1b in MySQL
2. Drop table question11 if it already exists in database h1b
3. Create table question11 in h1b database
4. Using Sqoop export command load the result from hdfs file format into question11 table in h1b database in MySQL
5. Display the result from question11 table

**CONCLUSION**

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

* + - 1. Sqoop we can use when we have data in SQL tables that we needs to be imported into Hadoop filesystem
      2. Hive helps in cleaning up of data and perform various SQL like queries on big data.
      3. For normal group by, join, filter based data retrieval Pig is very efficient.
      4. MapReduce code written in Java makes complex analysis easy. Codes required to be written to collect user inputs and complex join operations are handled efficiently using this approach.

**WEBOGRAPHY**

<https://hadoop.apache.org/docs/stable/hadoop-project-dist/hadoop-hdfs/HdfsDesign.html>

<https://www.edureka.co/blog/hadoop-ecosystem>

\*