



Capstone – Spring – 2017

Big Data Analysis By using Apache Hadoop

Srikanth Gottimukkula

Advisor: Professor Greg Gogolin

MISI 799



**ABSTRACT:**

Big data analytics is the process of analysis the extremely large sets of data. Analyzing the big data is a challenging task to the data analytics, because it is not viable to store the huge data on a traditional data warehouse which makes more expensive, which involves large distributed file system. Hadoop is an open source java based programming framework which is used in companies like Yahoo, Facebook, Twitter etc. to store and process the huge data sets by using the commodity hardware. Hadoop is created as faster alternative to the traditional RDBMS. In Traditional RDBMS is took lot of time to process the large data, by using Hadoop we can process huge data very fast. These Big data is not possible to analysis by the traditional data analytics. This project will present a study of store the big data and analysis by using Hadoop MapReduce and Apache Hive.

**Keywords:** Big data, Hadoop, MapReduce and Hive.

TABLE OF CONTENT

1. Proposal

1.1 Overview of Big Data----- 6

1.2 How Hadoop is useful in Big Data? ----- 6

1.3 Purpose of the Project ----- 6

1.4 Statement of the project ----- 7

1.5 Research Questions----- 7

1.6 Limitations ----- 7

1.7 Timeline ----- 8

1.8 Glossary ----- 9

2. Literature Review:

2.1 The Evolution of Big Data ----- 11

2.2 Benefits of Using Big Data ----- 12

2.3 Big Data Challenges. ----- 12

2.4 Big Data Solutions -----14

2.5 Hadoop Overview ----- 16

2.6 Key Factors of Hadoop ----- 17

2.7 Hadoop Distributed file System (HDFS) ----- 18

2.8 MapReduce ----- 19

2.9 Apache Hive ----- 20

3. Methodology

3.1 Research Questions ----- 21

**4. Findings**

4.1 Movie lens Dataset Schema ----- 26

4.2 Loading Data into HDFS ----- 26

4.3 Creating Table and Loading data into Hive ----- 27

4.4 Queries in Hive ----- 28

**5. References**

**List of Figures**

	<b>Page</b>
Fig -1: Categories of Big Data -----	11
Fig -2: Traditional approach for big data solution -----	14
Fig -3: Google approach for big data solution -----	15
Fig -4: Hadoop Framework -----	16
Fig -5: Apache Hive architecture -----	24
Fig -6: Load Data into HDFS -----	26
Fig 7: Creating tables in Apache Hive -----	27
Fig 8: Loading data into Hive table -----	28
Fig 9: Result for effect of gender on ratings. -----	29
Fig 10: Result for effect of Occupation in ratings. -----	29
Fig 11: Result for effect of age in ratings. -----	30
Fig 12: Word count program execution on eclipse. -----	34
Fig 13: Result for Word count program execution on eclipse. -----	34

## **CHAPTER-1 PROPOSAL**

### **1.1 Overview of Big Data:**

The term big data is referred as the data with large and complex datasets which is not possible to processing through traditional data processing techniques. The main challenges with the big data which includes data storage, data analysis, visualization, querying, and update and privacy of the data. The big data is not only in bigger in size of data which includes many techniques, various tools to processing the data and frameworks. Big data describes the amount of data which have both in structured and unstructured manner which is generated day to day business, it is important to know which data is useful for their organization. Big data helps to analysis the in-depth concepts of the data for the organization to take the decision making for develop their organization.

### **1.2 How Hadoop is useful in Big Data?**

In Big Data technology, Hadoop is a changing the technology in perception of handling the big data which is especially in handling unstructured data. By using simple processing model, the Apache Hadoop can handle the surplus data to be processed across the cluster of computer. In Apache Hadoop is made to scale up from the single machine to large cluster which consist of number of machines by offering a storage capacity and fast processing the data. Instead of depending on the hardware of the machine, Apache Hadoop provide high availability by using their library to detect and handle breakdown at the application layer of the data.

### **1.3 Purpose of the Project:**

The main purpose of this project is to handle the big data by using Hadoop technology. This project deals with the store the big data and process the data by using MapReduce and Hive

technologies. The objective is to load the Movie lens dataset in to Hadoop by using Hadoop Distribution File System and analysis the data by using HIVE technology and Load the Text file in to HDFS and process the text file by using Programming language called MapReduce.

## 1.4 Statement of the Problem:

The technology of Apache Hadoop is one of the most advanced technology in Big Data in most of the organizations like Facebook, Twitter, and Yahoo etc. With the rapid growth of the data in organization both in structured and unstructured data, Apache Hadoop is the solution for store and analysis the data with less cost for storage and fast processing time when compared with other technologies.

## 1.5 Research Questions:

1. How Hadoop is useful rather than the traditional data warehouse?
2. Features of using Hive QL rather than SQL in analysis of Movie lens data set?
3. How can HIVE QL be used to analysis the movie lens dataset in Hadoop?
4. How can Map Reduce be used for text mining in Hadoop?

## 1.6 Limitations:

Every project have some limitations, this project also have limitations and constraints.

The following are the limitation and constraints of the project:

1. **Time:** The project deadline is on 22<sup>nd</sup> April 2017 and project presentation is on 29<sup>th</sup> April 2017. The total duration of the project has about three months for completion which is less time to complete the project documentation and the presentation because of this new technology.

2. **Virtual Machine:** The project requires virtual machine to setup the Hadoop, there are many organization provide virtual machines to practice, like Horton Works, Cloudera with Ubuntu Linux version and CentoOS. By using virtual machines it take more time to processing and install the Hadoop framework on it.
3. **New Technology:** Hadoop is new technology is difficult to understand the ecosystem and their uses with in less time.
4. Due to limited resources from the open source, the project limited to only Apache HIVE and MapReduce.

**1.7 Timeline:**

For each and every project there is a timeline to complete the project in deadline and this project has no any exception. This project has seven major task to complete the project in scheduled time. The table below will explain the tasks that are needed to complete with dates that are estimated and description of tasks.

<b>Task</b>	<b>Estimated Time to Complete</b>	<b>Description</b>
Project Proposal	1/24/2017	Project Proposal consist of project overview, Research questions, limitations and glossary
Methodology	1/31/2017	This mainly consists of methods to complete the project like Hadoop installation in virtual Machine.
Implementation of Project	2/21/2017	Gathered all the requirement to execute the project.



Document finding	2/28/2017	Loaded all the data into Hadoop and get ready for processing the data.
Literature Review	3/15/2017	This Task consists of collection of all the previous technologies and Over view of the Hadoop technology
Project Findings	3/30/2017	Documentation of finding the research question is done.
Final Documentation	4/15/2017	Final Project documentation is done for review
Final Presentation	4/28/2017	The project presentation is ready for the presentation.

### 1.8 Glossary:

**Data Node:** It contain the all the data in the Hadoop in types of blocks.

**HDFS:** Hadoop Distributed File System, which breaks the large data into small blocks that are replicated and distributed across the cluster.

**Hive:** It is a data warehouse platform which is built on top of Hadoop for Data analysis and Querying. It is also useful to query data by using SQL like language called HQL.

**HQL (HiveQL):** SQL Like language used to query the data and execute MapReduce jobs in top of HDFS.

**Name Node:** It is main core of the HDFS File system which contained the Meta data of all files in the Cluster.

**YARN:** It is a resource manager for Hadoop 2.0 it helps to give the Meta data to the job tracker.

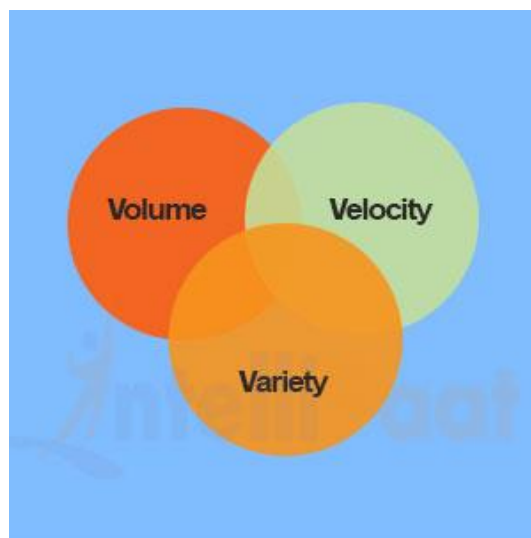
Aberration of YARN is “Yet another resource negotiator”

**MapReduce:** It is a software platform to parallel processing the data on large cluster of commodity hardware with reliable and fault tolerant.

## CHAPTER -2 LITERATURE REVIEW

### 2.1 The Evolution of Big Data:

In early 2000's Doug Laney (Industry analyst) defined the big data in to three categories as flows velocity, variety, and Volume.



**Fig -1 Categories of Big Data**

#### **Volume:**

The term big data, name itself refers the size of the data which is huge size. Every organizations collects the data from their relative sources like Social media, Business transactions and information gets from their other machines. Storing the big data was a big issues in olden days, but the new technology (such as Hadoop) has reduced the issues to store the huge data.

**Velocity:**

The term velocity refers to the speed at which the data will flow from the source to their business processes, like application logs, social media sites, etc. The flow of the data is massive and continuous the data will generate.

**Variety:**

In earlier days, spreadsheets and databases are the only sources of data which is used for many organizations. But now a days the data is unstructured which is in the form of images, Videos, audio and emails which is being considered for the analysis of data.

**Types of data**

Structured Data: Relational Data

Semi Structured Data: XML Data

Unstructured Data: Text, media logs, PDF etc.

**2.3 Benefits of Using Big Data:**

- Using the information in social media like Facebook and twitter the marketing companies are known about their company's response of their products and promotions.
- In real time analysis, big data is useful to find the root cause of failure, issues and defects.
- In the field of advertising, it is easy to know about the customer habit of buying their products.
- The implementation of big data tools are expensive, but in case of the organization can save a lot of money. It also reduce the cost of IT landscape to their organizations.

- Frauds can be detected at the moment it happens and proper security measures can be taken easily by the big data analytics.
- Dramatically increases the service of the organization because the older data will be available at all the time when the analysts need to analysis.

**Big Data Challenges:**

The analysis of big data consists of many phases which include data mining, data storage, data transfer, data integration and query processing. In each phase big data has many challenges. Scalability, Data quality, Cost Management and Security are the certain challenges of big data.

**Scalability:**

Managing a large and complex data which increasing rapidly in volume of data is a challenging issue in big data. Storing and analysis the large data with traditional software tools are not enough to manage the large volume of data. For example, Amazon web service, the average time taken to processes the 200 Node cluster is 4 minutes. It is not possible to processes with traditional software tools with in less time.

**Data Quality:**

The Data Quality is not a new concern, but to manage the big data and ability to store big data without any dirty data which means user input errors, duplicate data and incorrect data which is not any useful for the development of the organization.

**Cost Management:**

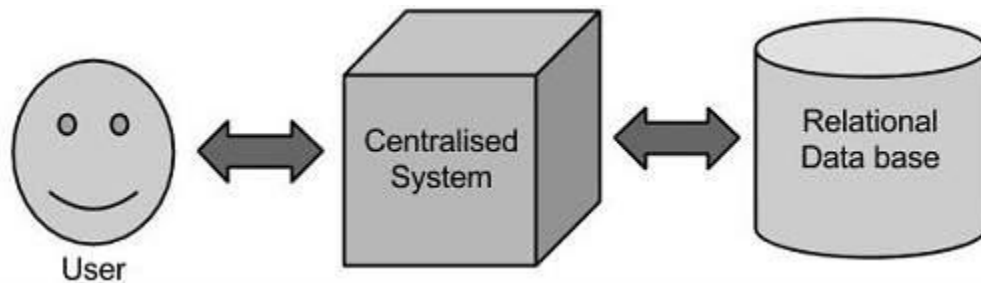
It is difficult for every organization to store the historical data, it required more hardware drive is reflects to high cost. The challenge lies in taking into account that the all cost of the project they required new drive space, to hire more professionals, to paying the cloud provider. Mainly in big data the cloud provider should carefully evaluate the usage of the disk space to determine the usage bill without any additional fee.

**Big Data solutions:**

There different approaches to face the big data challenges in that traditional approach, google solution and Hadoop are the main approaches.

**Traditional Approach:**

In this approach, the data will store and process in a Relational Database like Oracle Database, Microsoft SQL server or DB2 servers and some of the software which interact with these servers and process the required data which is useful for the analysis.



**Fig -2: Traditional approach for big data solution**

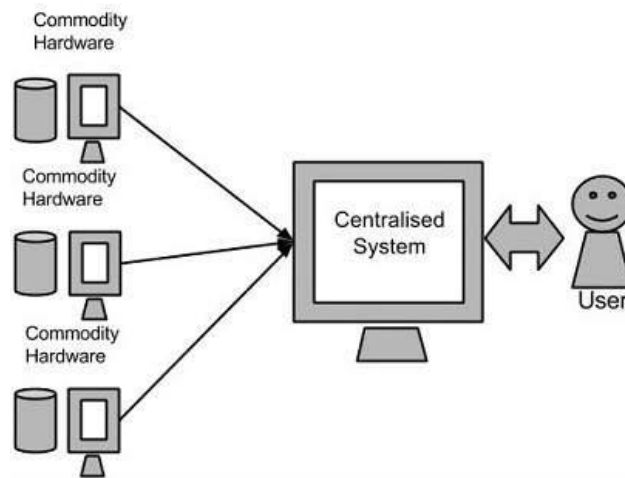
**Limitations:**

This traditional approach will be used for small volume of data can be process, when it comes to large and complex data the processing time is more. The source data should be

structured data, but recent days the data is in different format like text files, log files, photos and videos.

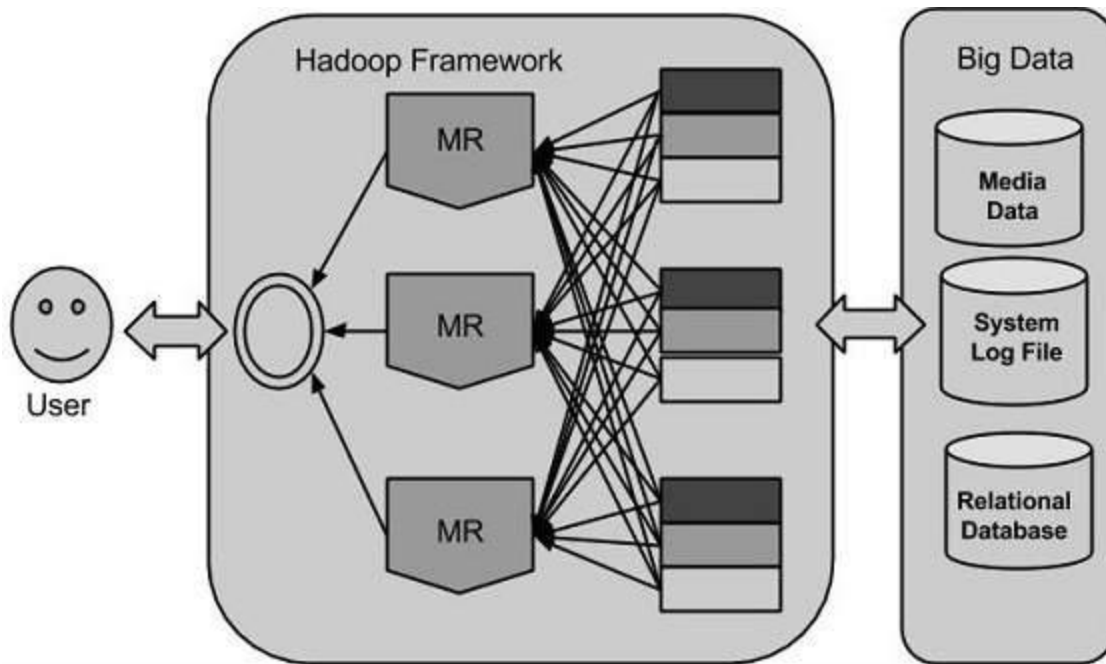
**Google Solution:**

Google find the solution for the traditional way of approach, by using a algorithm called MapReduce. This algorithm divide the task into small parts and assign the different small parts into different drives which is connected through network and collects the result in a dataset.



**Fig -3: Google approach for big data solution**

From this google solutions Doug Cutting, started an open source project called Hadoop which is now registered trademark for the Apache software. Hadoop runs application by using the google algorithm, where data is stored in commodity hardware and the data processed in parallel in cluster.



**Fig -4: Hadoop Framework**

### **Hadoop:**

Hadoop is open source framework which is used to store and processing the large or complex data in a distribution manner on a large cluster of commodity hardware. The main tasks for Hadoop is Store massive data storage and fast processing the data. It is one of the most important technology in big data.

### **Hadoop Architecture:**

Hadoop Framework consist of Four Modules as follows;

### **Hadoop Common:**

These consists of Java libraries and utilities which is required for the other Hadoop Modules. These libraries consists of required Java files to start the Hadoop.

### **Hadoop YARN:**



It is a resources management platform which is responsible for the job schedule for managing the compute resources in a cluster.

**Hadoop Distributed File System (HDFS):**

Distributed file system that helps to store the extremely large data set on a commodity hardware.

**Hadoop MapReduce:**

It is a parallel processing model for processing the large amount of data (peta bytes of data), on a large cluster with reliable and fault tolerant manner.

**The Key Factor of Hadoop:**

**Low Cost:** Hadoop is a free open Source framework work, to store significant amount of information in commodity hardware. Hadoop moreover offers a down to earth stockpiling answer for organization impacting data sets. The issue with standard social database organization systems is that it is incredibly incurred significant injury prohibitive to scale to such a degree to process such large volumes of data. With a ultimate objective to decrease costs, various organization in the past would have expected to down-illustration data and portray it in perspective of particular assumptions as to which data was the most critical.

**Fast Processing:** Hadoop has a unique method to store the data in a distributed method on the cluster. By using the Hadoop is effectively process terabytes of data in a just minute and petabytes of data in hour.

**Disadvantages:**

- Hadoop is not suitable for small business with less amount of data have.

- In Hadoop basic java programming language is used, without having any experience in java this chance for hacking.
- There are lot of stability problems in Hadoop.

**Hadoop Distributed File System (HDFS):**

HDFS is very fault-tolerant and is intended to be deployed on inexpensive hardware. HDFS provides high processing access to application knowledge and is appropriate for applications that have giant knowledge sets. Developed specifically for large scale processing workloads wherever measurability, flexibility and throughout square measure important, HDFS accepts knowledge in any format in spite of schema, optimizes for prime information measure streaming, and scales to provide deployments of 100PB and on the far side.

**Key features of HDFS:**

1. Availability: Serve mission important workflows and applications.
2. Fault Tolerance: mechanically get over failures.
3. Flexible Access: Multiple and open frameworks for publication and classification system amounts.
4. Load Balancing: place knowledge showing intelligence for max potency and utilization.
5. Replication: It gives multiple copies of every file provide knowledge protection and procedure performance.
6. Scale-Out Architecture: we will add Servers to extend capability.

**MapReduce:**

Map Reduce is a programming model associated with an implementation for process and generating giant data sets with a parallel, distributed formula on a cluster. Conceptually similar approaches are okay far-famed since 1995 with the Message Passing Interface normal having scale back and scatter operations. The term Map Reduce refers to two tasks where as separate and distinct tasks that Hadoop programs perform. The primary is that the map job, that takes a group of information and converts it into another set of information, wherever individual components square measure counteracted into tuples (key/value pairs). The Reduce job takes the output from a map as input and combines those knowledge tuples into a smaller set of tuples, because the sequence of the name MapReduce implies, the job is usually performed once the map job.

**Key Features of MapReduce:**

**Simplicity:** Developers will write applications in their language of selection, like Java, C++ or python, and map scale back jobs square measure simple to run.

**Scalability:** MapReduce will method peta bytes of information, hold on in HDFS on the cluster.

**Speed:** Parallel processing means, it will take issues that wont to take recently to resolve and solve them in hours or minutes.

**Recovery:** It takes care of failures. If a machine with one copy of the info is out of available, another machine incorporates a copy of identical key/value try, which might be wont to solve identical sub-task. The Job tracker keeps all the tasks in track

**Apache Hive:**

Apache Hive is knowledge warehouse infrastructure designed on high of Hadoop for providing knowledge summarization, query, and analysis. Whereas at first developed by Facebook, Apache Hive is currently used and developed by different corporations like Netflix. Amazon maintains a package fork of Apache Hive that's enclosed in Amazon Elastic Map scale back on Amazon Web Services.

## CHAPTER- 3 METHODOLOGY

## Research Questions:

## 1. How Hadoop is useful rather than the traditional data warehouse?

Characteristics	RDBMS	HADOOP
Basic Understanding	Traditional databases have only row-column databases which is used for transactional systems, reporting.	It is an open source framework to store data in a distributed file system on commodity hardware without any regular structured data.
Manufacturers	Microsoft SQL Server, MySQL, Oracle, Etc.	It is implemented by Cloudera, Hortonworks, Amazon AWS
Best for Applications	Traditional database used for only reads & write the reasonable data sets (Store less than GB)	Less cost when compared with the traditional databases and also store large datasets, structured and semi-structured and un structured data. (Store more than petabytes).
Scalability	Challenging to Scale –out process	It has a strong bias to open source community and java based program

Strength and Weakness	Massive data sets, does not allow semi structured and un structured data	Complex data sets, Code – based program, incompatible approach in organizations , writes (one at a time)
-----------------------	--	--

**2. Features of using Hive QL rather than SQL in analysis of Movie lens data set?**

SQL is a traditional language which is used for both transactional and analytical database. In HIVE is used for both transactional and analytical data which is more focused on analytical focus because in Hive no use of update and delete function. But in Hive use for fast processing of huge volume of data than SQL.

When is it best to use of Apache Hive?

Big data organizations want to process the data for fast analysis of data which is collected in over a period of time. Hive is an excellent tool for fast processing the data and analytical querying the historical data. In both SQL and Hive the structured data will processing. Hive is not suitable for the small organization where the data is very small. In Big data Hive might not be the best procedure but it helps to process the data very fast. Facebook is the one of the organization which is utilized the Hive for the real time analytics.

When is it best to use of SQL?

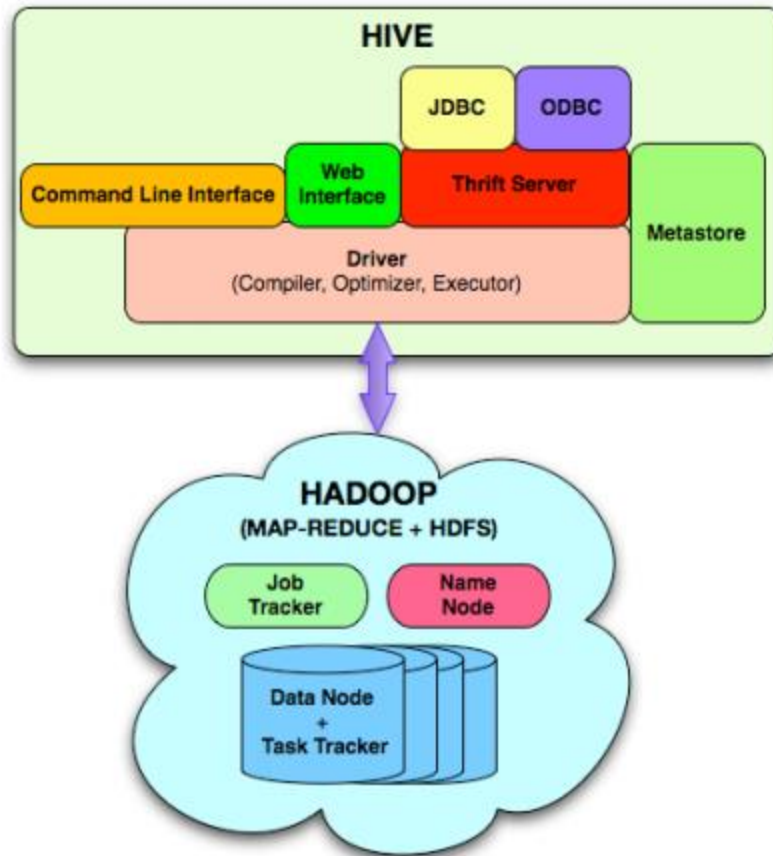
SQL is the most traditional information examination alternative among the three and its capacity to refresh itself in accordance with developing client desires make it applicable even today. While it is unquestionably a superior choice than exceed expectations for genuine information investigators, it misses the mark when business requests quick paced preparing and

examination. Be that as it may, when the necessities are not very requesting, SQL is an amazing tool. Its nature and adaptability discover support with designers. Given that an extensive segment of engineer group the world over knows about SQL, its utilization makes them gainful from the very first moment. It gives engineers the office to develop and upgrade it, which make it very adjustable.

### **3. How can HIVE QL be used to analysis the movie lens dataset in Hadoop?**

For the end user who may have no idea about Java Programming and MapReduce programming, Hive is the best solution and interesting project because it allows the important and best parts of the Hadoop in both MapReduce and Data Storage. When compared with traditional method of querying like SQL, Hive summaries the big data and make easy way of querying and analysis the big data. Hive is data warehouse infrastructure tool in top of Hadoop which process the structured data. HiveQL is the desired language supported by HIVE which is same like as SQL. These HiveQL Query complied into the MapReduce Job and then executed the command in Hadoop Cluster. In addition to these HQL enables the series of MapReduce jobs into queries. Hive keeps the metadata in a relational database model to support the features of the data.

The Basic architecture of the proposed system of the dataset are as follows



**Fig -5: Apache Hive architecture**

#### 4. How can Map Reduce be used for text mining in Hadoop?

In Hadoop, MapReduce is a Java programming that decompose the large data processing job into small individual tasks that can be executed in parallel processing across the cluster. The Result of all the individual task will be combined into a single output result.

MapReduce consists of two main functions Map and Reduce. In Map function, the data converts into a set of individual data set, where the individual data broken into tuple (key value, Value pair) For Example: For this project use word count program where the output of the Map function will be like (Word, 1). In Reduce function, reduce function takes the output of the Map function as an input and combines the data in tuples into smaller set (Key value, value pair).



The workflow of MapReduce in big data processing consists of five steps they are Splitting, Mapping, Intermediate Splitting, Reduce and Combiner.

**Splitting:** The Splitting parameter can be anything like comma, space, tab. In a dataset, MapReduce split the data into small dataset separated by comma or space, Tab or even with new line.

**Mapping:** In Mapping, dataset converts the output into (Key value, value pair) format.

**Intermediate Splitting:** In entire process each mapper job will perform in different machine in a cluster. In order to combine the result of the each mapper job the group the same key pair into a same cluster.

**Reducer:** In this stage the output of the mapper will group in phase.

**Combining:** This is the last phase where the individual data from the each cluster will be combined in the form of output.

## CHAPTER 4 RESEARCH FINDINGS

### 4.1 Movie lens Dataset Schema:

For the analysis, the dataset is chosen from the <http://grouplens.org/datasets/movielens> download the 1 million movie lens data set and stored the data file in HDFS. The dataset consists of four files Ratings, Movies, User and Occupation.

Ratings [ user\_id, movie\_id, rating, Timestamp]

User [ user\_id, Gender, Age, Occupation, Zip\_code ]

Movies: [Movie-id, Title, Genres]

Occupation[ Occupation\_id, occupation].

### 4.2 Loading data in to HDFS:

For the above mentioned four tables Ratings, movie, user, occupation all the tables are loaded in to HDFS. The schema for loading data in to HDFS are shown below fig

**Syntax:** Hadoop fs -put capstone/movielens /user/training/capstone

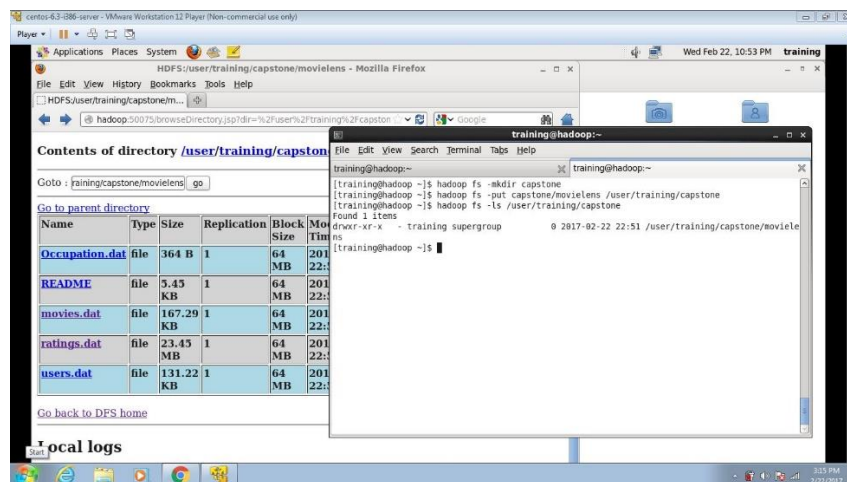


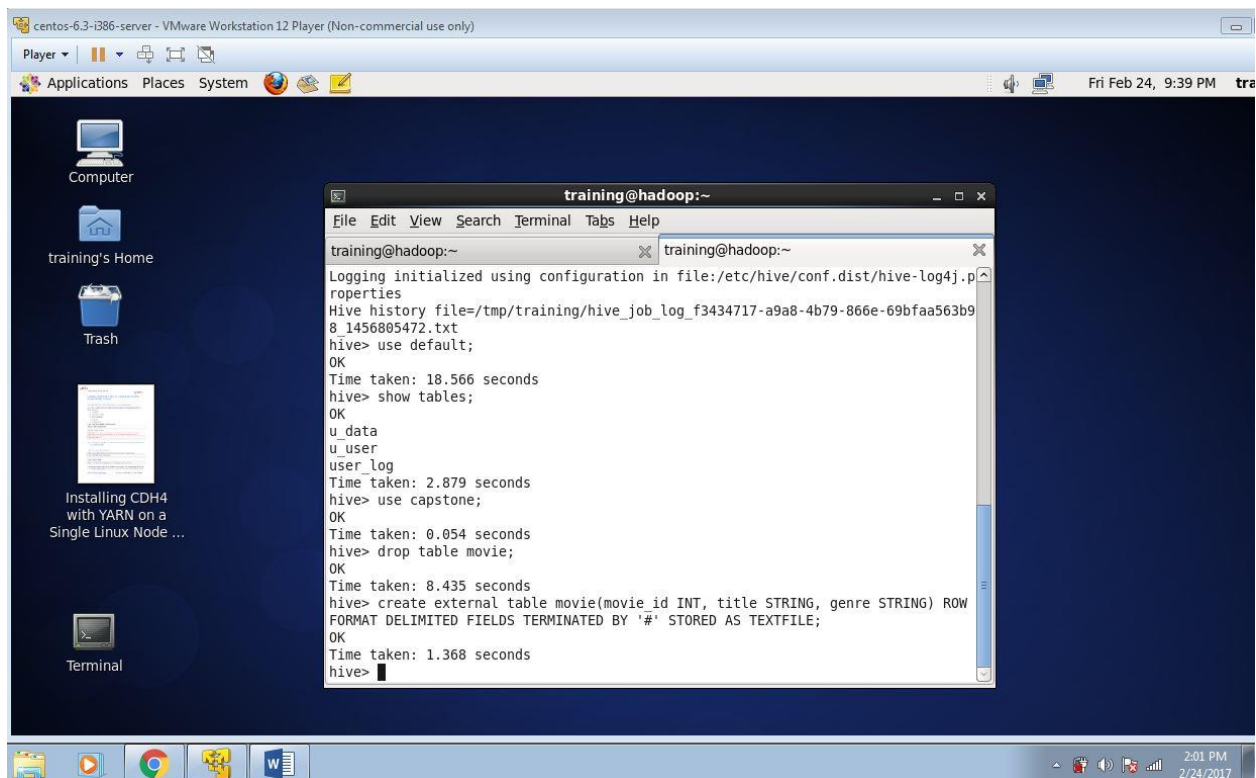
Fig -6: Load Data into HDFS

### 4.3 Creating tables and loading data on HIVE:

For the mentioned four files user, movie, ratings and occupation tables are stored in HDFS. Tables are created at HDFS by using Hive QL, which is similar as SQL query.

#### Syntax:

```
CREATE EXTERNAL TABLE MOVIE ( movie_id INT, title STRING, genre STRING ) ROW
FORMAT DELIMITED FIELDS TERMINATED BY '#' STORED AS TEXTFILE;
```



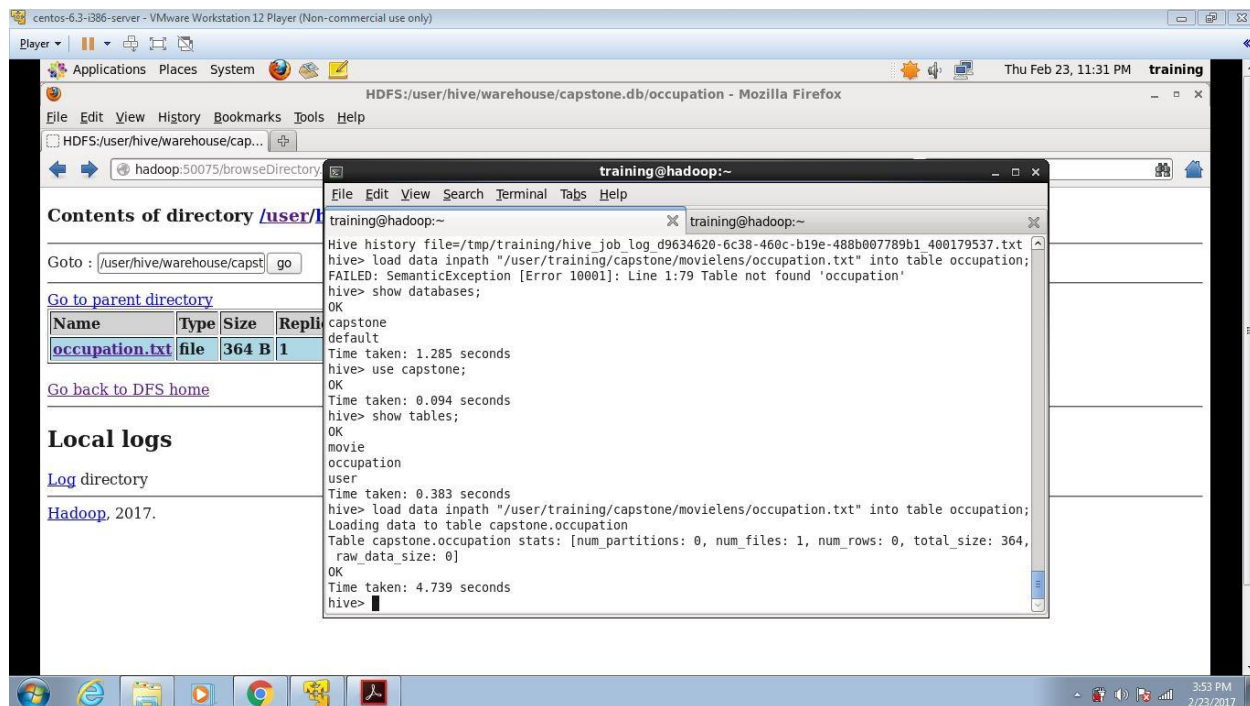
**Fig 7: Creating tables in Apache Hive**

The next step is to load the data in to tables after creating all the four tables.

#### Syntax:

```
LOAD DATA INPATH <file path > INTO TABLE <table_name>
```

LOAD DATA INPATH "/user/training/capstone/movielens/movies.txt" INTO TABLE movie;



**Fig 8: Loading data into Hive table.**

### 4.3 Queries in Hive :

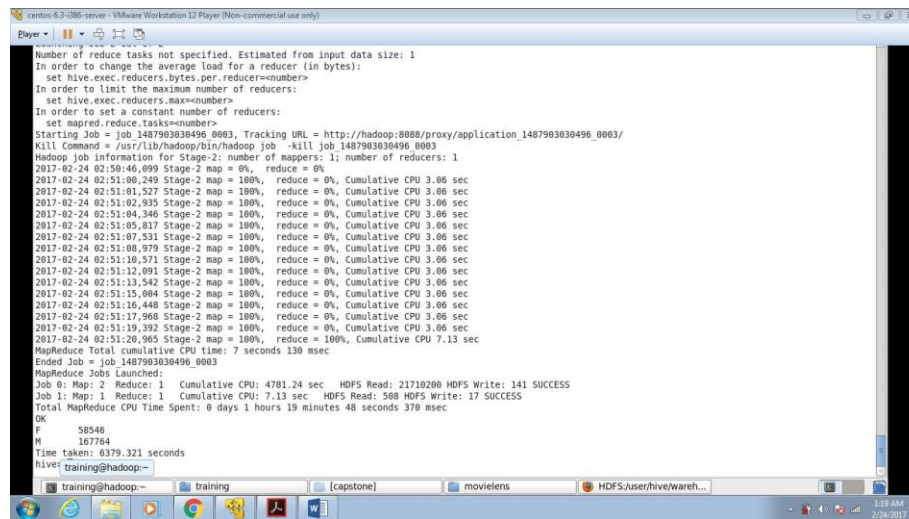
#### 4.3.1 Effect of Gender on the ratings.

The effect of ratings that the users gave to the movies which group by gender. The HiveQL query to get the effect of user's gender with the rating 5.

```
SELECT u.gender, COUNT(*)
FROM rating r
INNER JOIN user u ON r.user_id = u.user_id
AND r.rating = 5
GROUP BY gender;
```

**RESULT:**

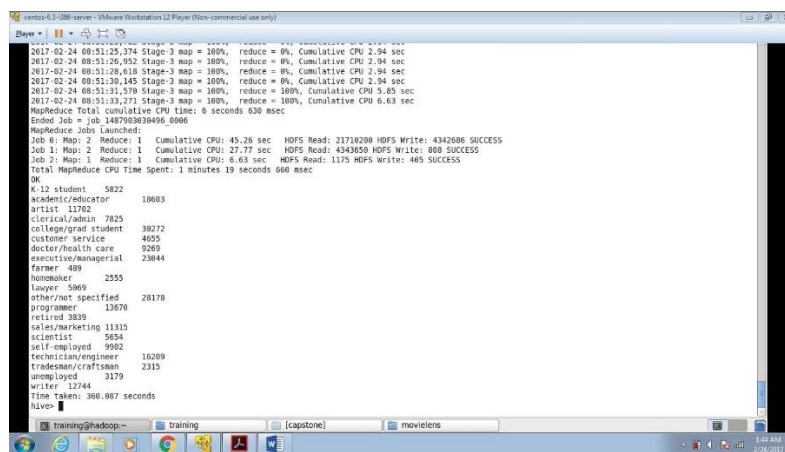
F	58546
M	167764



**Fig 9: Result for effect of gender on ratings.**

**4.3.2 Effect of occupation in ratings.**

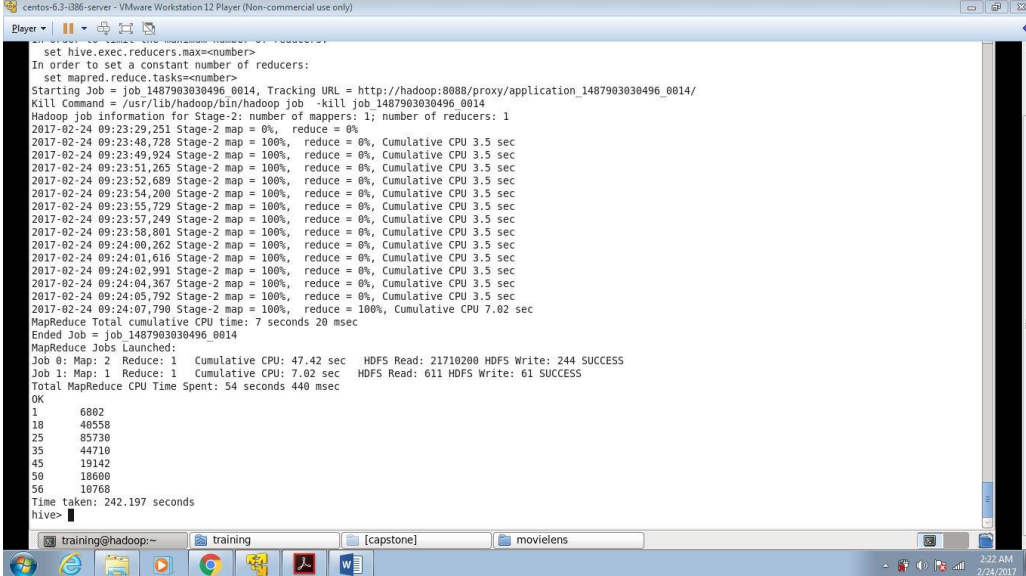
SELECT o.name, count(\*) FROM movie rating r, occupation o, user u WHERE r.userid = u.id AND o.id = u.occupationid AND r.rating = 5 GROUP BY o.name;



**Fig 10: Result for effect of Occupation in ratings.**

### 4.3.3. Effect of age on ratings.

```
SELECT u.age, count(*) FROM movie rating r, user u WHERE r.userid = u.id AND
r.rating = 5 GROUP BY u.age;
```



```
centos-6.3-086-server - VMware Workstation 12 Player (Non-commercial use only)
Player
set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
set mapred.reduce.tasks=<number>
Starting Job = job_1487903030496_0014, Tracking URL = http://hadoop:8080/proxy/application_1487903030496_0014/
Kill Command = /usr/lib/hadoop/bin/hadoop job -kill job_1487903030496_0014
Hadoop job information for Stage-2: number of mappers: 1; number of reducers: 1
2017-02-24 09:23:29,251 Stage-2 map = 0%, reduce = 0%
2017-02-24 09:23:48,728 Stage-2 map = 100%, reduce = 0%, Cumulative CPU 3.5 sec
2017-02-24 09:23:49,924 Stage-2 map = 100%, reduce = 0%, Cumulative CPU 3.5 sec
2017-02-24 09:23:51,265 Stage-2 map = 100%, reduce = 0%, Cumulative CPU 3.5 sec
2017-02-24 09:23:52,689 Stage-2 map = 100%, reduce = 0%, Cumulative CPU 3.5 sec
2017-02-24 09:23:54,200 Stage-2 map = 100%, reduce = 0%, Cumulative CPU 3.5 sec
2017-02-24 09:23:55,729 Stage-2 map = 100%, reduce = 0%, Cumulative CPU 3.5 sec
2017-02-24 09:23:57,249 Stage-2 map = 100%, reduce = 0%, Cumulative CPU 3.5 sec
2017-02-24 09:23:58,801 Stage-2 map = 100%, reduce = 0%, Cumulative CPU 3.5 sec
2017-02-24 09:24:00,262 Stage-2 map = 100%, reduce = 0%, Cumulative CPU 3.5 sec
2017-02-24 09:24:01,616 Stage-2 map = 100%, reduce = 0%, Cumulative CPU 3.5 sec
2017-02-24 09:24:02,991 Stage-2 map = 100%, reduce = 0%, Cumulative CPU 3.5 sec
2017-02-24 09:24:04,367 Stage-2 map = 100%, reduce = 0%, Cumulative CPU 3.5 sec
2017-02-24 09:24:05,792 Stage-2 map = 100%, reduce = 0%, Cumulative CPU 3.5 sec
2017-02-24 09:24:07,790 Stage-2 map = 100%, reduce = 100%, Cumulative CPU 7.02 sec
MapReduce Total cumulative CPU time: 7 seconds 20 msec
Ended Job = job_1487903030496_0014
MapReduce Jobs Launched:
Job 0: Map: 2 Reduce: 1 Cumulative CPU: 47.42 sec HDFS Read: 21710200 HDFS Write: 244 SUCCESS
Job 1: Map: 1 Reduce: 1 Cumulative CPU: 7.02 sec HDFS Read: 611 HDFS Write: 61 SUCCESS
Total MapReduce CPU Time Spent: 54 seconds 440 msec
OK
1 6802
18 40558
25 85730
35 44710
45 19142
50 18600
56 10760
Time taken: 242.197 seconds
hive>
```

Fig 11: Result for effect of age in ratings.

### Text Analysis by Using MapReduce Program:

Word Count Program:

package PackageDemo;

import java.io.IOException;

import org.apache.hadoop.conf.Configuration;

import org.apache.hadoop.fs.Path;

import org.apache.hadoop.io.IntWritable;

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

import org.apache.hadoop.io.Text;

import org.apache.hadoop.mapreduce.Job;

import org.apache.hadoop.mapreduce.Mapper;

import org.apache.hadoop.mapreduce.Reducer;

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

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

import org.apache.hadoop.util.GenericOptionsParser;

public class WordCount {

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

{

Configuration c=new Configuration();

String[] files=new GenericOptionsParser(c,args).getRemainingArgs();

Path input=newPath(files[0]);

Path output=newPath(files[1]);

Job j=new Job(c,"wordcount");

j.setJarByClass( WordCount.class);

j.setMapperClass (MapForWordCount.class);
```

```
j.setReducerClass (ReduceForWordCount.class);

j.setOutputKeyClass (Text.class);

j.setOutputValueClass (IntWritable.class);

FileInputFormat.addInputPath (j, input);

FileOutputFormat.setOutputPath (j, output);

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

}

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

public void map(LongWritable key, Text value, Context con) throws IOException,
InterruptedException

{

String line = value.toString();

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

for(String word: words )

{

    Text outputKey = new Text(word.toUpperCase().trim());

    IntWritable outputValue = new IntWritable(1);

    con.write(outputKey, outputValue);
```



```
}}}  
  
public static class ReduceForWordCount extends Reducer<Text, IntWritable, Text, IntWritable>  
  
{  
  
public void reduce(Text word, Iterable<IntWritable> values, Context con) throws IOException,  
InterruptedException  
  
{  
  
int sum = 0;  
  
    for(IntWritable value : values)  
  
    {  
  
sum += value.get();  
  
    }  
  
    con.write(word, new IntWritable(sum));  
  
}}}
```

The above program consists of three classes Driver class, Mapper Class, and Reducer Class.

The Below images shows the execution of the word count program

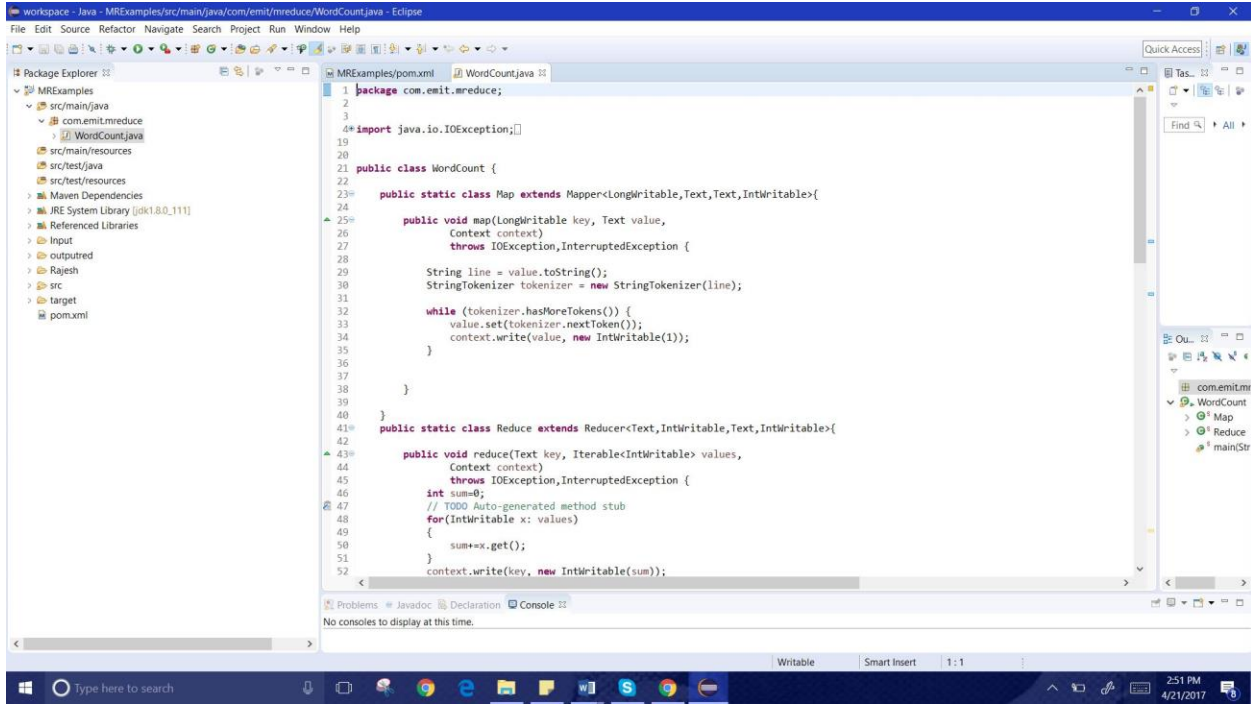


Fig 12: Word count program execution on eclipse.

Result:

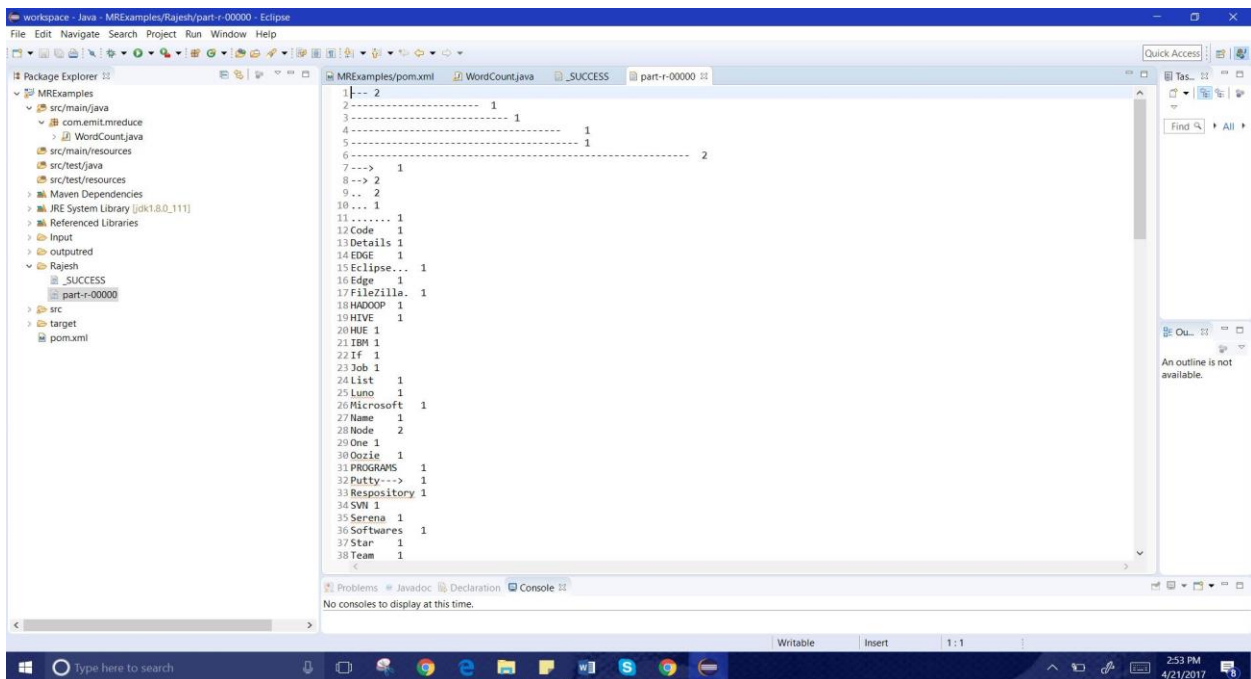


Fig 13: Result for Word count program execution on eclipse.

**Conclusion:**

The conclusion on this project is that we have executed and explored different in queries execution on Hive for extensive datasets. Mapping and decreasing functionalities of MapReduce also, HDFS hived to handle bigger and unstructured datasets. Executing MapReduce work raises the sorts and size of examination that can be connected to large datasets, which is impractical by other customary handling framework. Hadoop can be utilized to take care of an assortment of analysis issues more efficiently. The realities which were uncovered amid the procedure can be utilized for building up some expectation models. Hive is less advanced when contrasted with conventional databases like Oracle, MySQL, and PostgreSQL so enhancement in Hive has great research conceivable outcomes. Additionally, Hive does not bolster Update and Delete usefulness yet. So research can be advance in this area. The facts which were uncovered amid the procedure can be utilized for building up some expectation models.

**Reference**

White, T. (2012). *Hadoop: the definitive guide*. Beijing: O'Reilly.

Allouche, G. (2015, July 01). Hadoop 101: An Explanation of the Hadoop Ecosystem - DZone Big Data. Retrieved April 27, 2017, from <https://dzone.com/articles/hadoop-101-explanation-hadoop>

Braselton, J. P. (2014). *Hadoop: integration in IBM, Microsoft and SAS*. Place of publication not identified: CreateSpace,2014

Seshachala, S. (2015, June 01). Bigdata - Understanding Hadoop and Its Ecosystem. Retrieved April 27, 2017, from <https://devops.com/bigdata-understanding-hadoop-ecosystem/>

Teplow, D. (2015, May 15). Hadoop . Retrieved April 27, 2017, from <http://www.hadoop360.com/blog/hadoop-whose-to-choose>

Mehta, S. (2016, June). Hadoop Ecosystem: An Introduction . Retrieved from <https://www.ijsr.net/archive/v5i6/NOV164121.pdf>

Kanoje, S. (2016, July 12). Hadoop Ecosystem Quick Start: 5 Key Components. Retrieved April 27, 2017, from <https://www.ironsidegroup.com/2015/12/01/hadoop-ecosystkey-components/>

Karambelkar, H. (2015). *Scaling big data with Hadoop and Solr: understand, design, build, and optimize your big data search engine with Hadoop and Apache Solr*. Birmingham: Packt Pub.

Prajapati, V. (2013). *Big Data Analytics with R and Hadoop*. Olton: Packt Publishing. Retrieved from <http://ebookcentral.proquest.com.ezproxy.ferris.edu/lib/ferrisstate/detail.action?docID=1477486>

Trifu, M. R., & Ivan, M. (2016). Big data components for business process optimization. *Informatica Economica*, 20(1), 72-78.  
doi:<http://dx.doi.org.ezproxy.ferris.edu/10.12948/issn14531305/20.1.2016.07>

Laura IVAN, M. (2016, January). *Big Data Components for Business Process Optimization*. Retrieved April 27, 2017, from <http://revistaie.ase.ro/content/77/07%20-%20Trifu,%20Ivan.pdf>

Prasad Padhy , R. (2013, February). *Big Data Processing with Hadoop-MapReduce in Cloud Systems* . Retrieved April 27, 2017, from <http://www.iaesjournal.com/online/index.php/IJ-CLOSER/article/view/1508/502>