# MOVIE ANALYSING SYSTEM

**A Project Report Submitted**
**In Partial Fulfillment of the Requirements**
**for the Degree of**

## MASTER OF COMPUTER APPLICATION

**by**

**AISHWARYA MITTAL**
**(1900290149005)**

**Under the Supervision of**

**Dr. Ajay Kumar Shrivastava**
**KIET Group of Institutions, Ghaziabad**

**to the**

**FACULTY OF MCA**

**DR. APJ ABDUL KALAM TECHNICAL UNIVERSITY**
**(Formerly Uttar Pradesh Technical University) LUCKNOW**

**July 2021**

# DECLARATION

I hereby declare that the work presented in this report entitled "MOVIE ANALYSING SYSTEM'', was carried out by me. I have not submitted the matter embodied in this report for the award of any other degree or diploma of any other University or Institute.

I have given due credit to the original authors/sources for all the words, ideas, diagrams, graphics, computer programs, experiments, results, that are not my original contribution. I have used quotation marks to identify verbatim sentences and given credit to the original authors/sources.

I affirm that no portion of my work is plagiarized, and the experiments and results reported in the report are not manipulated. In the event of a complaint of plagiarism and the manipulation of the experiments and results, I shall be fully responsible and answerable.

Name : Aishwarya Mittal

Roll. No. : 1900290149005

Branch : Master of Computer Applications

# CERTIFICATE

Certified that **Aishwarya Mittal** (**1900290149005**) has carried out the project work presented in this report entitled "**Movie Analysing System**" for the award of **Master of Computer Application** from Dr. A.P.J. Abdul Kalam Technical University, Lucknow under my supervision. The report embodies result of original work, and studies are carried out by the student himself and the contents of the report do not form the basis for the award of any other degree to the candidate or to anybody else from this or any other University.

**Dr. Ajay Kumar Shrivastava** **External Examiner**

Professor & Head

Dept. of Computer Applications

KIET Group of Institutions, Ghaziabad

Date:

# CERTIFICATE

This is to certify that the project report entitled "***Movie Analysing System***" submitted to KIET Group of Institutions in partial fulfillment of the requirement for the award of the degree of **MASTER OF COMPUTER APPLICATION**, is an original work carried out by **Ms. Aishwarya Mittal,** University Roll No: **1900290149005** under the guidance of Dr. Ajay Kumar Shrivastava.

The matter embodied in this project is a genuine work done by the student and has not been submitted whether to this University or to any other University / Institute for the fulfillment of the requirement of any course of study.

Dr. Ajay Kr. Shrivastava
(Head – CA)
KIET, Ghaziabad

# ABSTRACT

In today's world there is a huge growth in data. This data is generated from variety of sources like social media, industry, transaction records, cell phone, GPS signals etc. It is difficult and challenging to store such a huge amount data in traditional data warehouse. Big Data is the dataset with 3 V's that are Volume, Variety and Velocity and difficult to store and process using traditional database management systems. Big Data Analytics is the way of processing the large amount of data. Hadoop is a popular open source software which is very useful in analyzing the larger data. Hadoop provides several tools for this purpose like Hive, Pig, Hbase, Cassandra etc. In this project we have used Hive with Hadoop framework for analysing movie dataset. Hive is built on the top of Hadoop and it has its own query language HiveQL which is similar to SQL. Hive will internally convert the queries into MapReduce job. The reason for why Hive is better than MySQL is that, Hive is most suitable for larger datasets and MySQL is suitable for smaller datasets. We have got significant improvement in processing time for analyzing dataset compared to traditional system.

# ACKNOWLEDGEMENT

Success in life is never attained single handedly. My deepest gratitude goes to my Project supervisor, **Dr. Ajay Kumar Shrivastava, Professor and Head, Department of Computer Applications** for his guidance, help and encouragement throughout my research work. Their enlightening ideas, comments, and suggestions. Words are not enough to express my gratitude for his insightful comments and administrative help at various occasions.

Fortunately, I have many understanding friends, who have helped me a lot on many critical conditions.

Finally, my sincere thanks go to my family members and all those who have directly and indirectly provided me moral support and other kind of help. Without their support, completion of this work would not have been possible in time. They keep my life filled with enjoyment and happiness.

**Aishwarya Mittal**
**1900290149005**

# TABLE OF CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

# CHAPTER 1

# INTRODUCTION

## 1.1 PROJECT DESCRIPTION

Today the data is growing in a very high speed. These data can be produced by any sources like industries, social media, cell-phones, scientific source etc. We can refer this large data as Big Data. Till now there is no particular measure is defined on size of the Big Data. In the beginning, Big Data was adopted by Facebook, LinkedIn, Google etc. The reason might be the rapid change of the data. The three characteristics of Big Data are volume, velocity and variety. Volume refers to the size of the data. It is growing bigger day by day. According to the expert's analysis, few years later the data can cross 25 Zettabytes. Velocity refers to how fast the data is being processed. For this we can consider the examples like posting comment or image on Facebook and watching video on YouTube etc. Variety is referring to different types of data and different sources which produce these data. The data can be structured, semi-structured or unstructured. It can be in any formats like image, csv files, text files, audio, video etc. In addition to these characteristics, two more have been defined: veracity and value. Veracity refers to the trustworthiness of the data and value refers to extracting meaningful information from datasets.

As the data are produced, we need to store and analyse it. The traditional data processing techniques are failed to analyse the Big Data. Today about 80% of data are unstructured and these unstructured data are impossible to analyse using relational database systems. So Big Data analytics is introduced for processing these large amount of structured, unstructured and semi-structured data with the help of existing software tools and with very less amount of time. Big data analytics makes use of Hadoop framework for analyzing the larger datasets. Hadoop is an open-source, reliable, scalable and shared computing framework developed especially to process the huge amount of data. It was initially developed by Google in 2004 and at present it is maintained by Apache. The main two components Hadoop are Hadoop Distributed File System (HDFS) and MapReduce. HDFS stores the data in the form of blocks. MapReduce is used for processing, sharing and clustering.

Hadoop ecosystem provides several tools for Big Data analytics. Some of the tools are Hive, Pig, Hbase, Cassandra, Mahout, Flume, Avro etc.

Pig is an analysis tool of Hadoop ecosystem. Pig has its own programming language called Pig Latin. To covert the Pig Latin scripts into MapReduce job Pig Runtime is used. Hive is called as data warehousing software. It has its own query language called HiveQL. It is used for processing the larger datasets stored indatawarehouse. Hbase is used to store the large amount of data especially structured data. It stores data in the form of tables and Pig or Hive queries can be used to analyse these data. Like Hbase, Cassandra is also a database which stores the structured data. In Casandra, the data is replicated in several nodes so even if the one node fails it doesn't make any difference. Its data will be available in other nodes. ThereforeCassandra is called as fault-tolerant system. Hbase and Cassndra are NoSQL databases and they are column oriented. Mahout is a framework developed by Apache. The main purpose of this is to implement the data mining algorithms. Flume is another data analyzing tool provided by Hadoop eco-system. It is used especially to analyse the log data. Its architecture is very simple and it is robust. Avro is schema-dependent. It is mainly used to serialise the data so that it can analysed easily. It declares different data structures with the help of JSON format. Java, C, C++, Python, C# and Ruby are the languages supported by Avro.

## 1.2 BIG DATA AND ITS CHARACTERISTICS

**Big Data** is a collection of data that is huge in volume yet growing exponentially with time. It is a data with so large size and complexity that none of traditional data management tools can store it or process it efficiently. Big data is also a data but with huge size.

In recent years, Big Data was defined by the "*3Vs*" but now there is "*5Vs*" of Big Data which are also termed as the characteristics of Big Data as follows:

**1. Volume:**

- The name 'Big Data' itself is related to a size which is enormous.
- Volume is a huge amount of data.
- To determine the value of data, size of data plays a very crucial role. If the volume of data is very large then it is actually considered as a 'Big Data'. This means whether a particular data can actually be considered as a Big Data or not, is dependent upon the volume of data.
- Hence while dealing with Big Data it is necessary to consider a characteristic 'Volume'.
- *Example:* In the year 2016, the estimated global mobile traffic was 6.2 Exabytes(6.2 billion GB) per month. Also, by the year 2020 we will have almost 40000 ExaBytes of data.

**2. Velocity:**

- Velocity refers to the high speed of accumulation of data.
- In Big Data velocity data flows in from sources like machines, networks, social media, mobile phones etc.
- There is a massive and continuous flow of data. This determines the potential of data that how fast the data is generated and processed to meet the demands.
- Sampling data can help in dealing with the issue like 'velocity'.
- *Example:* There are more than 3.5 billion searches per day are made on Google. Also, FaceBook users are increasing by 22%(Approx.) year by year.

**3. Variety:**

- It refers to nature of data that is structured, semi-structured and unstructured data.
- It also refers to heterogeneous sources.
- Variety is basically the arrival of data from new sources that are both inside and outside of an enterprise. It can be structured, semi-structured and unstructured.
    - **Structured data**: This data is basically an organized data. It generally refers to data that has defined the length and format of data.
    - **Semi- Structured data**: This data is basically a semi-organised data. It is generally a form of data that do not conform to the formal structure of data. Log files are the examples of this type of data.
    - **Unstructured data**: This data basically refers to unorganized data. It generally refers to data that doesn't fit neatly into the traditional row and column structure of the relational database. Texts, pictures, videos etc. are the examples of unstructured data which can't be stored in the form of rows and columns.

**4. Veracity:**

- It refers to inconsistencies and uncertainty in data, that is data which is available can sometimes get messy and quality and accuracy are difficult to control.
- Big Data is also variable because of the multitude of data dimensions resulting from multiple disparate data types and sources.
- *Example:* Data in bulk could create confusion whereas less amount of data could convey half or Incomplete Information.

**5. Value:**

- After having the 4 V's into account there comes one more V which stands for Value!. The bulk of Data having no Value is of no good to the company, unless you turn it into something useful.
- Data in itself is of no use or importance but it needs to be converted into

something valuable to extract Information. Hence, you can state that Value! is the most important V of all the 5V's.

## 1.3 BIG DATA ANALYSIS USING HADOOP

Hadoop is a framework that allows you to first store Big Data in a distributed environment, so that, you can process it parallelly. There are basically two components in Hadoop. The first one is ***HDFS*** for storage (Hadoop distributed File System), that allows you to store data of various formats across a cluster. The second one is ***YARN***, for resource management in Hadoop. It allows parallel processing over the data, i.e. stored across HDFS.

- **HDFS**: HDFS creates an abstraction, let me simplify it for you. Similar as virtualization, you can see HDFS logically as a single unit for storing Big Data, but actually you are storing your data across multiple nodes in a distributed fashion. HDFS follows master-slave architecture. In HDFS, Namenode is the master node and Datanodes are the slaves. Namenode contains the metadata about the data stored in Data nodes, such as which data block is stored in which data node, where are the replications of the data block kept etc. The actual data is stored in Data Nodes.I also want to add, we actually replicate the data blocks present in Data Nodes, and the default replication factor is 3. Since we are using commodity hardware and we know the failure rate of these hardwares are pretty high, so if one of the DataNodes fails, HDFS will still have the copy of those lost data blocks.

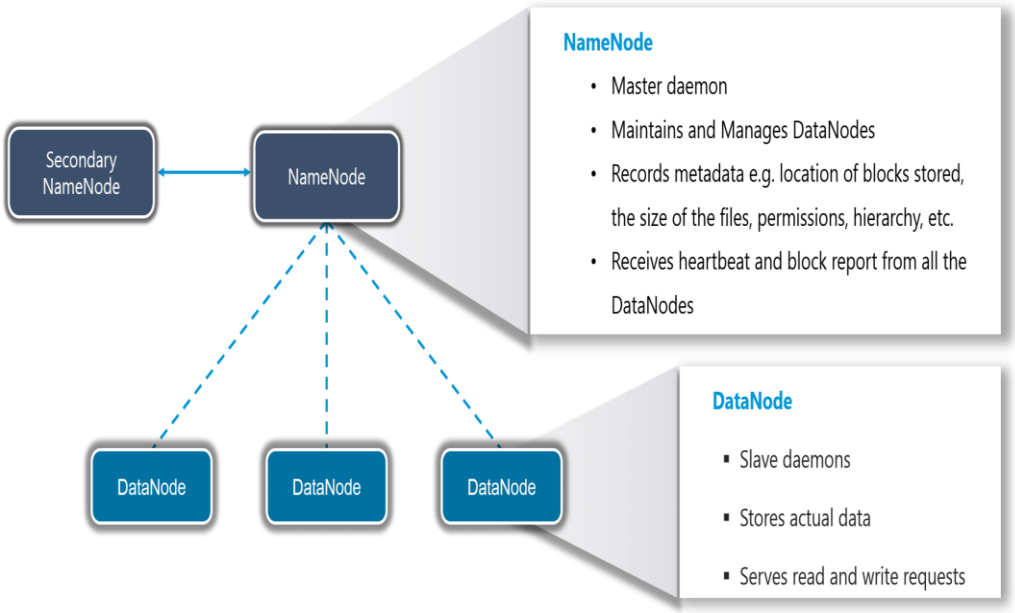



Fig. 1.1 HADOOP-HDFS

- **YARN**: **YARN** performs all your processing activities by allocating resources and scheduling tasks.It has two major components, i.e. ResourceManager and NodeManager.

ResourceManager is again a master node. It receives the processing requests and then passes the parts of requests to corresponding NodeManagers accordingly, where the actual processing takes place. NodeManagers are installed on every DataNode. It is responsible for the execution of the task on every single DataNode.

- **MapReduce** — **MapReduce** is both a programming model and big data processing engine used for the parallel processing of large data sets. Hadoop MapReduce is a software framework for easily writing applications which process vast amounts of data (multi-terabyte data-sets) in-parallel on large clusters (thousands of nodes) of commodity hardware in a reliable, fault-tolerant manner.

A MapReduce *job* usually splits the input data-set into independent chunks which are processed by the *map tasks* in a completely parallel manner. The framework sorts the outputs of the maps, which are then input to the *reduce tasks*. Typically both the input and the output of the job are stored in a file-system. The framework takes care of scheduling tasks, monitoring them and re-executes the failed tasks.

- **Hadoop Common** — Hadoop Common provides a set of services across libraries and utilities to support the other Hadoop modules.

As big data grows exponentially, parallel processing capabilities of a Hadoop cluster help in increasing the speed of analysis process. However, the processing power of a hadoop cluster might become inadequate with increasing volume of data. In such scenarios, hadoop clusters can scaled out easily to keep up with speed of analysis by adding extra cluster nodes without having to make modifications to the application logic.

A hadoop cluster architecture consists of a data centre, rack and the node that actually executes the jobs. Data centre consists of the racks and racks consists of nodes. A medium to large cluster consists of a two or three level hadoop cluster architecture that is built with rack mounted servers. Every rack of servers is interconnected through 1 gigabyte of Ethernet (1 GigE). Each rack level switch in a hadoop cluster is connected to a cluster level switch which are in turn connected to other cluster level switches or they uplink to other switching infrastructure.

In a single node hadoop cluster, all the daemons i.e. DataNode, NameNode, TaskTracker and JobTracker run on the same machine/host. In a single node hadoop cluster setup everything runs on a single JVM instance. The hadoop user need not make any configuration settings except for setting the JAVA_HOME variable. For any single node hadoop cluster setup the default replication factor is one.

In a multi-node hadoop cluster, all the essential daemons are up and run on different machines/hosts. A multi-node hadoop cluster setup has a master slave

architecture where in one machine acts as a master that runs the NameNode daemon while the other machines acts as slave or worker nodes to run other hadoop daemons. Usually in a multi-node hadoop cluster there are cheaper machines (commodity computers) that run the TaskTracker and DataNode daemons while other services are run on powerful servers. For a multi-node hadoop cluster, machines or computers can be present in any location irrespective of the location of the physical server.

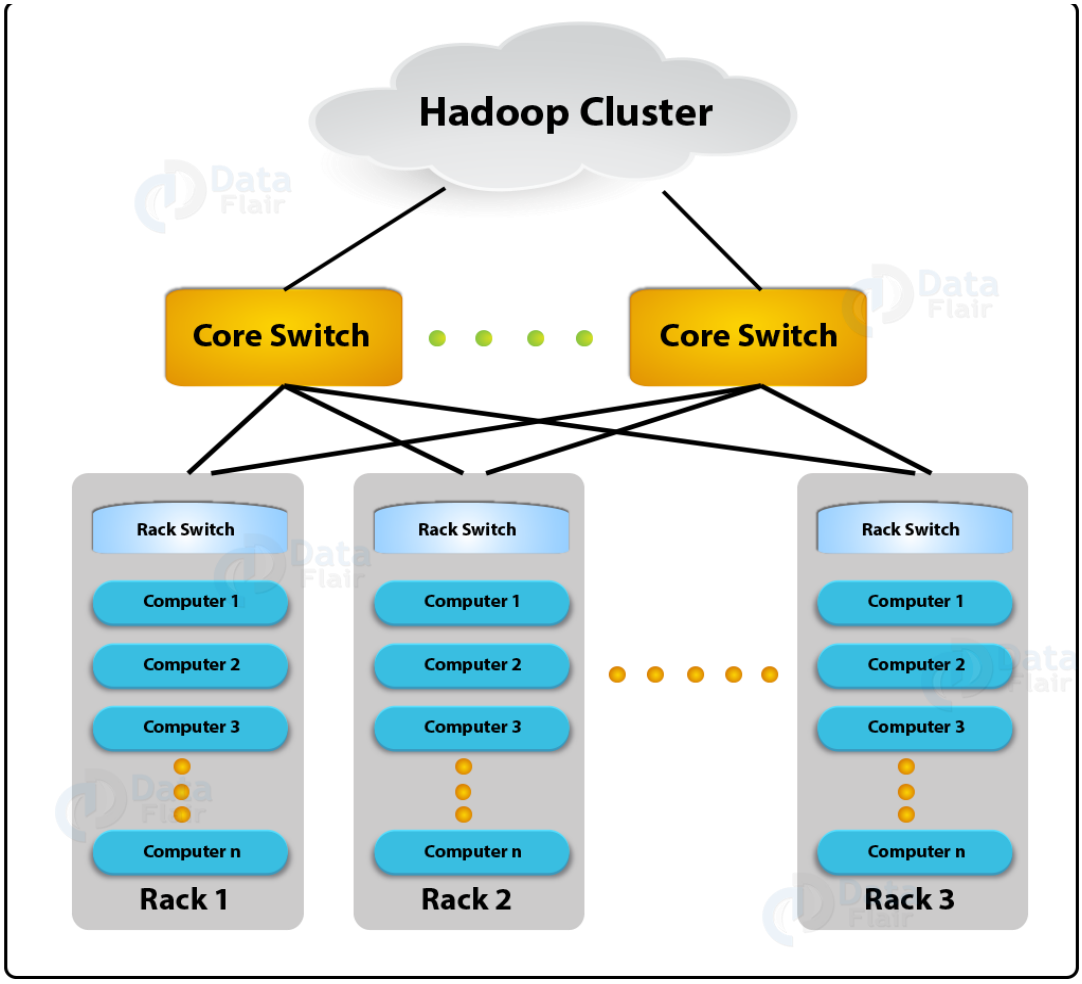



Fig. 1.2 HADOOP CLUSTER

# CHAPTER 2

# LITERATURE REVIEW

## 2.1 BIG DATA

Big Data is the dataset with 3 V's that are Volume, Variety and Velocity and difficult to store and process using traditional database management systems. Big Data Analytics is the way of processing the large amount of data. Hadoop is a popular open-source software which is very useful in analyzing the larger data. Hadoop provides several tools for this purpose like Hive, Pig, Hbase, Cassandra etc. [16]

Advances in information technology and its widespread growth in several areas of business, engineering, medical, and scientific studies are resulting in information/data explosion. Knowledge discovery and decision-making from such rapidly growing voluminous data are a challenging task in terms of data organization and processing, which is an emerging trend known as *big data computing*, a new paradigm that combines large-scale compute, new data-intensive techniques, and mathematical models to build data analytics. Big data computing demands a huge storage and computing for data curation and processing that could be delivered from on-premise or clouds infrastructures.[6]

With the development of Internet of Things (IoT), 5 G, and cloud computing technologies, the amount of data from manufacturing systems has been increasing rapidly. With massive industrial data, achievements beyond expectations have been made in the product design, manufacturing, and maintain process. [12]Many research works deal with big data platforms looking forward to data science and analytics. These are complex and usually distributed environments, composed of several systems and tools. As expected, there is a need for a closer look at performance issues.[14]

## 2.2 HADOOP

Hadoop is Java based programming framework for distributed storage and processing of large data sets on commodity hardware. It is developed by Apache Software Foundation as open source framework. Hadoop basically has two main components. First one is Hadoop Distributed File System (HDFS) for distributed storage and second part is MapReduce for distributed processing. HDFS is a file system which builds on the existing file system. MapReduce is a programming model which is used for processing and generating large data sets with a parallel, distributed algorithm on a cluster. A MapReduce job generally splits the input data set into independent blocks which are processed by the map tasks in a completely parallel manner. First step is mapping of data set in MapReduce architecture. The framework sorts the outputs of the mapping process, which are then input to the second step is reduce task. Input and the output of the job are stored in a file-system. [3]

In Hadoop, users retain control over how data are being processed by writing their own algorithms in the Map and Reduce interfaces provided. Other issues that require addressing when dealing with parallel executions, such as failure detection, recovery and synchronization between tasks, are handled automatically by the Hadoop framework. In conjunction, the underlying file system supporting the Hadoop framework, the Hadoop Distributed File System, HDFS 2 provides the framework with features such as file distribution balancing and file redundancy. The replicated files not only provide data redundancy in the event where one of the nodes fails, it also helps Reduce data transfer when performing workload balancing. In cases where the nodes are busy, new tasks can be assigned to other nodes which hold the corresponding block's replicated copy, for processing. [5]

## 2.3 HDFS

HDFS faces several issues when it comes to handling a large number of small files. These issues are well addressed by archive systems, which combine small files into larger ones. They use index files to hold relevant information for retrieving a small file content from the big archive file. However, existing archive-based solutions require significant overheads when retrieving a file content since additional processing and I/Os are needed to acquire the retrieval information before accessing the actual file content, therefore, deteriorating the access efficiency.[9]

## 2.4 MAPREDUCE

MapReduce is an established computing paradigm for processing massive data, in which the input data is viewed as records of key-value pairs. In such a two-phase computation, a first-phase *map* task processes a portion of the data records to generate

or update the key-value pairs; these key-value pairs are then shuffled and supplied to the second-phase *reduce* tasks, each processes a portion of them, typically with the same key, to produce the final output. [10]

Over the last decade, several advancements have happened in distributed and parallel computing. A lot of data is generated daily from various sources, and this speedy data proliferation led to the development of many more frameworks that are efficient to handle such huge data e.g. - Microsoft Dryad, Apache Hadoop, etc. Apache Hadoop is an open-source application of Google MapReduce and is getting a lot of attention from various researchers. Proper scheduling of jobs needs to be done for better performance. [2]

[7] analyze the performance impact of JobTracker failure in Hadoop. A JobTracker failure is a serious problem that affects the overall job processing performance.

## 2.5 APACHE HIVE

SQL-on-Hadoop engines such as Hive provide a declarative interface for processing large-scale data over computing frameworks such as Hadoop.The increasing need to process analytical queries over large-scale semi-structured data has led to the development of SQL-on-Hadoop engines. These systems evaluate SQL-like queries over data stored in distributed file systems such as the Hadoop Distributed File System (HDFS) . Hive was the first SQL-on-Hadoop system to provide an SQL-like query language, namely HiveQL, and can use MapReduce or Tez as its underlying framework for executing queries. [8]

## 2.6 MOVIE ANALYSIS

Ammar Fuad et al. proposed a method to analyze the performance MySQL cluster, Hive and Pig [17]. For the experiment three different sizes of movie lens datasets are considered. The result showed that MySQL cluster processing time increases as the data size increases. Because of step-by-step execution nature of the Pig, its processing time exceeded the processing time of the Hive.

Karan Sachdeva et al. compared the performance of MapReduce, Hive and Pig by considering unstructured, semistructured and structured dataset[19]. From the result it's been proved that to process structured data Hive is the efficient tool. For processing semi-structured and unstructured data Pig and Map-Reduce respectively are efficient. Analysis of Meteorological and Oceanographic data is difficult using MySQL because it consists of several number of small files and each file might contain 20 to 300 columns. Ali Usman Abdullahi et al. have analysed the Meteorological and Oceanographic data for indexed and non-indexed table using Hive. There are three different types of queries

have been executed on the tables. It is possible to reduce the number of mappers so that response time can be increased.

Aditya Bhardwaj et al. have analysed Twitter data using Hive. Analysis is performed to predict the Map-Reduce time and total job completion time for different cluster size[20].

It is possible to increase the performance of Hadoop/Hive with help of Multi Query Optimization technique and distributed Hive. Varun Garg considered 11 queries from TPC-H and different sizes of datasets are used. From the experiments the author has shown that performance of distributed Hive is greater than the conventional Hive. Xiaoyu Wang et al. performed analysis on Internet traffic data using Hadoop and Hive based traffic analysis system. The libpcap files are pre-processed with less amount of time. They have shown that the system proposed by them is error free and increases execution speed. Taoying Liu et al. presented the implementation Standard Science DBMS which is a benchmark of distributed scientific data on Hive. Hive queries are compared with SciDB queries. The amount of time taken to load the data by both SciDB and Hive is same. For the smaller input data the performance of Hive is slower when compared to SciDB. S K Pushpa et al. analyzed airport data using Hive and found out that it is more efficient and faster when compared to traditional approach. Dharaben Patel et al. have analyzed the huge amount of network traffic data using Hive. Hive queries are written to find-out different types of security attacks. To visualize the result Apache Zeppelin tool is used. As the part of future work using this method more number of security attacks can be found. Hive performance time can be predicted by determining the Map-Reduce job execution time. Amit Sangroya et al. have proposed a linear regression model. Hive processing time decreases with increase in the size of data. The proposed method helps in predicting the Hive performance time with reduced error rate.[15]

The Connected Vehicles can exchange information about location and security. Large amount of data are produced by Connected Vehicle. Weija Xu et al. analyzed these data using Hive and compared result with PostgreSQL. The experiment conducted showed that Hive query performance time is lesser than PostgreSQL. The Earth science data are always in NetCDF format. This format is not supported in HDFS, therefore we cannot analyse this data using Haddop tools. Shujia Zhou et al. [18] proposed a system that will convert the NetCDF format to CSV format making it easy to visualize and to analyze by Haddop Tools like Spark, Hive.

S. Karimian-Aliabadi et al. continuous struggle of data scientists with increasing size of data to be analyzed, led to handful of practical tools and methods. In 2008, Dean and Ghemawat proposed MapReduce (MR) paradigm to process large amount of data on multiple node cluster to increase parallelism and therefor improved performance.[1]

The MR paradigm was not globally used until useful Hadoop framework

developed in 2011 by Apache. The Hadoop Distributed File System (HDFS) is a primary layer of the Hadoop ecosystem but not the only one. In 2013, Vavilapalli et al., introduced YARN layer to the Hadoop cluster in order to specialize the resource management and make it dynamic rather than Hadoop's earlier static allocation scheme.With more complex dataflow in MR applications there was a need to cut down the complexity into multiple stages and thus Directed Acyclic Graphs (DAG) was chosen by Tez developers to demonstrate the dataflow between stages of a complex application. Taking advantage of the memory's high speed and the Resilient Distribute (RDD) concept, Spark was created and became popular due to high speed and the ease of application development.[2]

Tuning the framework and cluster parameters in order to reduce the execution time of a BigData application was a challenge from the earliest steps and a main part of this optimization process is to predict the execution time for a given set of parameters. But With each step in development of a more advanced framework for processing BigData, new set of parameters and complexity is created and execution time prediction made more and more challenging. A lot of works have been done in literature to simulate, model, or learnthe process, but their accuracy and scalability is only enough for simple runs with a single job running by one or more users and not for more complex applications with multiple multi-stage jobs running by number of users.[3]

In today's world there is a huge growth in data. This data is generated from variety of sources like social media, industry, transaction records, cell phone, GPS signals etc. It is difficult and challenging to store such a huge amount data in traditional data warehouse. Big Data is the dataset with 3 V's that are Volume, Variety and Velocity and difficult to store and process using traditional database management systems. Big Data Analytics is the way of processing the large amount of data. Hadoop is a popular open source software which is very useful in analyzing the larger data. Hadoop provides several tools for this purpose like Hive, Pig, Hbase, Cassandra etc. In this paper, we have used Hadoop framework. For the analysis of movie dataset Hive tool is used with Hadoop framework. We have got significant improvement in processing time for analyzing dataset compared to traditional system.

HDFS needs to work with massive amounts of data stored in very large files. When dealing with large HDFS files, MapReduce splits the files into multiple pieces at record boundaries, so it can read data from the large file simultaneously by starting multiple mapper processes. A splittable data format lets a file be correctly split into pieces at the record boundaries. Hadoop environments prefer to use binary formats rather than text formats when dealing with HDFS, because binary formats prevent incomplete records being written to files, by catching and ignoring incorrect records that may be created due to data corruption or incompleteness. This type of issue can occur, for example, when a cluster accidentally runs out of space during a write. Data compression capability is also a key requirement for a good HDFS file format. A popular binary format used by many is the Avro container file format, which is

splittable and can also be compressed. Another common HDFS data format is a SequenceFile, which is a splittable file format represented as a list of keys and values. Users can also customize the data format by using serializers, which let them write data in any format they choose.

In paper, predictive models for the box office performance of the movies was represented by factors derived from social media and IMDb. According to our models, we have identified the following patterns: the popularity of leading actress is crucial to the success of a movie, the combination of past successful genre and a sequel movie isanother pattern for success, a new movie in the not popular genre and an actor with low popularity could be a pattern for a Flop. It is surprising that sentiment score and view and comment counts were not identified as relevant in our experiments. Author believe it is related to how weights are assigned to each attribute. Further studies to determine different weighting methods will be beneficial. In addition, our prediction is for movies yet to be released. The preliminary result of tracking 13 of the movies shows a good prediction performance from our model. A follow-up study on the final performance of our models will be validated and presented once all of the movies are released. Future work to improve our models will include further refinement of the Neutral class and characterization of movie box office performance in terms of net profits and profit ratios.

Apache Hive provides a SQL interface that enables you to use HDFS data without having to write programs using MapReduce. It's important to understand that unlike Apache HBase, Hive is not a database. It simply provides a mechanism to project a database structure on data you store in HDFS and lets you query that data using HiveQL, a SQL-like language. Hive uses a type of SQL that lets you query HDFS data in ways that are similar to how you query data stored in a relational database. While HiveQL doesn't have the full range of features available in SQL, it offers more than enough SQL capabilities for you to efficiently work with HDFS data. When you use a Hive query, Hive parses the SQL query and generates a MapReduce job to process the data to get you the query results. The main rationale for Hive is to reduce effort by doing away with developing MapReduce programs. It also provides a data warehouse capability when handling large amounts of data, is analyst friendly and is ideal for making use of HDFS data for business intelligence (BI) analysis.

Two mainstream approaches for large-scale data analysis are parallel database systems and MapReduce-based systems . Both approaches share certain common design elements: they both employ a shared-nothing architecture , and deployed on a cluster of independent nodes via a high-speed interconnecting network; both achieve parallelism by partitioning the data and processing the query in parallel on each partition. However, parallel database approach has major limitations on managing and querying spatial data at massive scale. Parallel database management systems (DBMSs) tend to reduce the I/O bottleneck through partitioning of data on multiple parallel disks and are not optimized for computational-intensive operations such as spatial and

geometric computations. Partitioned parallel DBMS architecture often lacks effective spatial partitioning to balance data and task loads across database partitions. While it is possible to induce a spatial partitioning, fixed grid tiling, for example, and map such partitioning to one dimensional attribute distribution key, such an approach fails to handle boundary objects for accurate query processing. Scaling out spatial queries through a parallel database infrastructure is possible while being costly, and such approach is explored.[4]

# CHAPTER 3

# REQUIREMENT GATHERING AND ANALYSIS

## 3.1 Hardware Requirements

- RAM 4GB – 8GB
- Operating system (32bit or 64 bit)
- 2 GHz or Higher Processor
- 20 GB Hard Disk

## 3.2 Software Requirements

- Oracle VM VirtualBox

Oracle VM VirtualBox is cross-platform virtualization software that allows users to extend their existing computer to run multiple operating systems at the same time. Designed for IT professionals and developers, Oracle VM VirtualBox runs on Microsoft Windows, Mac OS X, Linux, and Oracle Solaris systems and is ideal for testing, developing, demonstrating, and deploying solutions across multiple platforms on one machine.

Oracle VM VirtualBox has been designed to take advantage of the innovations introduced in the x86 hardware platform, and it is lightweight and easy to install and use. Yet under the simple exterior lies an extremely fast and powerful virtualization engine. With a well-earned reputation for speed and agility, Oracle VM VirtualBox contains innovative features to deliver tangible business benefits: excellent performance; a powerful virtualization system; and a wide range of supported guest operating system platforms.VirtualBox is used to setup the virtual hadoop servers.

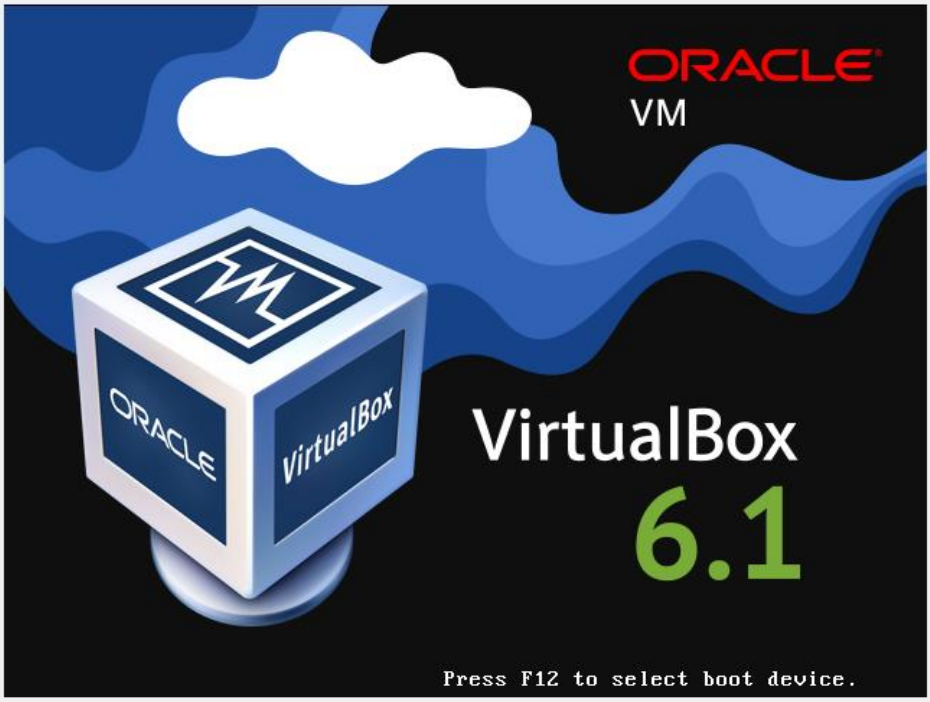



Fig. 3.1 ORACLE VIRTUALBOX

- CentOS 7

CentOS is a community-driven free software effort that provides two Linux distribution (CentOS Linux and CentOS Stream) and a variety of Special Interest Groups releasing packages to run on those distributions. CentOS Linux provides a free, community-supported computing platform functionally compatible with its upstream source, Red Hat Enterprise Linux (RHEL).CentOS Stream is a continuously delivered distribution that tracks just ahead of RHEL and acts as an upstream for RHEL development.

Fig. 3.2 CENTOS

- Apache NiFi version-1.9.0

Apache NiFi supports powerful and scalable directed graphs of data routing, transformation, and system mediation logic. Some of the high-level capabilities and objectives of Apache NiFi include:

a) Web-based user interface
   - Seamless experience between design, control, feedback, and monitoring

b) Highly configurable
   - Loss tolerant vs guaranteed delivery
   - Low latency vs high throughput
   - Dynamic prioritization
   - Flow can be modified at runtime

c) Data Provenance
   - Track dataflow from beginning to end

d) Designed for extension
   - Build your own processors and more
   - Enables rapid development and effective testing

- Apache Hadoop

  - Hadoop is an open source framework utilized for processing humungous datasets and also used for distributed storage.
  - A particular special type of computational cluster is built in order to store and analyze large volumes of unstructured data is known as a Hadoop cluster.
  - Hadoop clusters are gaining popularity for enhancing the speed of data analysis applications. Hadoop clusters are extremely scalable.
  - Hadoop clusters are highly efficient as they are resistant to failures.


Fig. 3.3 APACHE HADOOP

- Apache Hive

  - Hive is a data warehouse system for Hadoop.
  - It allows querying, data analysis utilizing HiveQLetc.
  - Hive enables users to potray structure on huge unstructured data.
  - Hive has the ability to understand organized and unorganized data which may include text files where fields are circumscribed by specific characters.

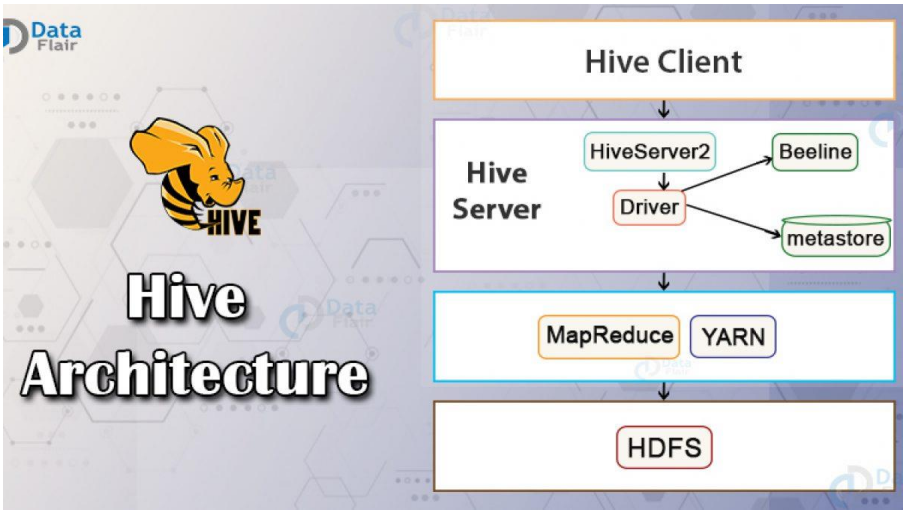


Fig. 3.4 HIVE ARCHITECHTURE

- Tableau

  Tableau is a powerful and fastest growing data visualization tool used in the Business Intelligence Industry. It helps in simplifying raw data in a very easily understandable format. Tableau helps create the data that can be understood by professionals at any level in an organization. It also allows non-technical users to create customized dashboards.
  The best features of Tableau software are-
  - Data Blending
  - Real time analysis
  - Collaboration of data

## 3.3 Data Requirements

The dataset requirement for our project is fulfilled through kaggle repository. From here we downloaded the dataset and used as an input. Kaggle.com is a website that provides dataset for free for its users. Thus we got dataset for free of cost.

# CHAPTER 4

# Design

## 4.1 Data Flow Diagram

- **0-level DFD**

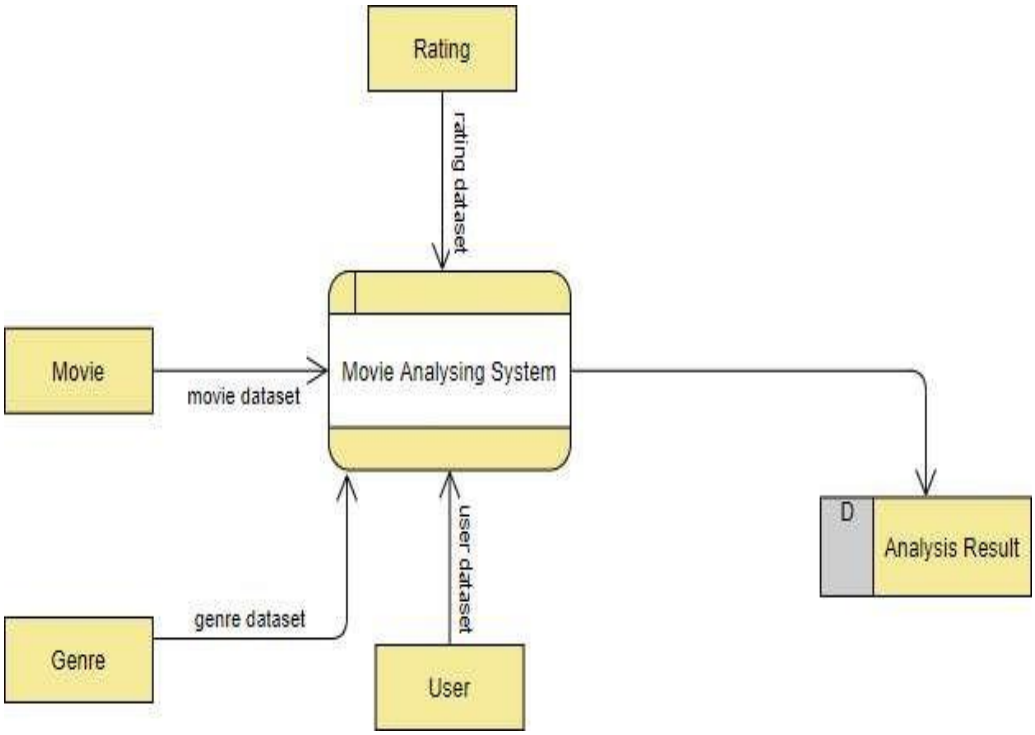



Figure: 4.1 0-level DFD

- **1-level DFD**

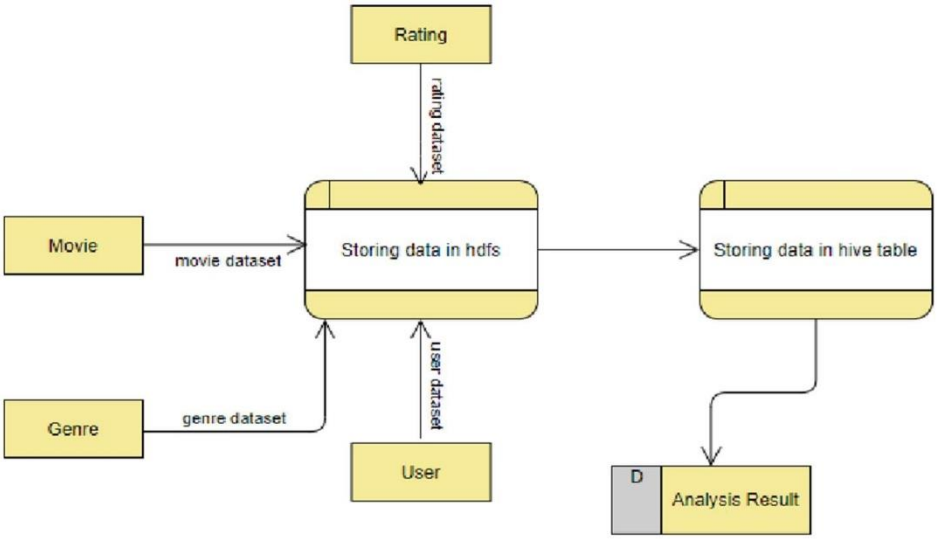



Figure: 4.2 1-level DFD

## 4.2 Data Dictionary

The datasets are taken from kaggle. The Movie dataset consists of information about the year of release, title, language, imdb ratings, FB likes, genre etc. In this report we have considered four datastes for analysis: Movie dataset, Genre dataset, Rating dataset, User dataset. Movie dataset consists of 10650 rows, Genre dataset consists of 5044 rows, Rating dataset consists of 6-7 lakh rows and User dataset consists of 6040 rows. The description about the datasets is as follows.

| | |
|---|---|
| MOVIE TITLE | Title of the movie. |
| LANGUAGE | Language used in movie. |
| YEAR | Movie release year. |
| IMDB RATINGS | IMDB ratings for a particular movie. |
| FB LIKES | Facebook likes obtained for a particular movie. |

Table 4.1 movie dataset

| | |
|---|---|
| USER ID | Unique ID for a particular user. |
| MOVIE ID | Unique ID for a particular movie. |
| RATINGS | The rating given by the user for a particular movie. |

Table 4.2 rating dataset

| | |
|---|---|
| MOVIE ID | Unique ID for a particular movie. |
| TITLE | Name of the movie. |
| GENRE | Describes the category to which a movie belongs. |

Table 4.3 genre dataset

| | |
|---|---|
| USER ID | Unique ID for a particular user. |
| GENDER | Describes whether user is male or female. |

Table 4.4 user dataset

# CHAPTER 5

# TESTING

## 5.1 BIG DATA TESTING STRATEGY

Big Data Testing is a testing process of a big data application to ensure that all the functionalities of a big data application work as expected. The goal of big data testing is to make sure that the big data system runs smoothly and error-free while maintaining the performance and security. Since big data is a collection of large datasets that cannot be processed using traditional computing techniques, traditional data testing methods do not apply to big data. This means your big data testing strategy should include big data testing techniques, big data testing methods and big data automation tools, such as Apache's Hadoop. There are several areas in Big Data where big data testing strategy is required. There are various types of testing in Big Data projects such as Database testing, Infrastructure, and Performance Testing, and Functional testing. Big Data defined as a large volume of data structured or unstructured. Data may exist in any format like flat files, images, videos, etc. The primary Big Data characteristics are three V's - Volume, Velocity, and Variety where volume represents the size of the data collected from various sources like sensors, transactions, velocity described as the speed (handle and process rates) and variety represents the formats of data. Some primary examples of Big Data are Social Networking sites like Twitter and Facebook and E-commerce sites such as Amazon, Flipkart, Snapdeal and any other E-commerce site which have millions of visitors and products.

Big Data Testing plays a vital role in Big Data Systems. If Big Data systems not appropriately tested, then it will affect business, and it will also become tough to understand the error, cause of the failure and where it occurs. Due to which finding the solution for the problem also becomes difficult. If Big Data Testing performed correctly, then it will prevent the wastage of resources in the future.

## 5.2 HOW BIG DATA TESTING STRATEGY WORKS

- **Data Ingestion Testing:** In this, data collected from multiple sources such as CSV, sensors, logs, social media, etc. and further, store it into HDFS. In this testing, the primary motive is to verify that the data adequately extracted and correctly loaded into HDFS or not. Tester must ensure that the data properly ingests according to the defined schema and also have to verify that there is no data corruption. The tester validates the correctness of data by taking some little sample source data, and after ingestion, compares both source data and ingested data with each other. And further, data loaded into HDFS into desired locations.

- **Data Processing Testing:** In this type of testing, the primary focus is on aggregated data. Whenever the ingested data processes, validate whether the business logic is implemented correctly or not. And further, validate it by comparing the output files with input files. **Tools -** Hadoop, Hive, Pig, Oozie

- **Data Storage Testing:** The output stored in HDFS or any other warehouse. The tester verifies the output data correctly loaded into the warehouse by comparing the output data with the warehouse data. **Tools -** HDFS, HBase

- **Data Migration Testing:** Majorly, the need for Data Migration is only when an application moved to a different server or if there is any technology change. So basically, data migration is a process where the entire data of the user migrated from the old system to the new system. Data Migration testing is a process of migration from the old system to the new system with minimal downtime, with no data loss. For smooth migration (elimination defects), it is essential to carry out Data Migration testing.

## 5.3 HOW TO TEST HADOOP APPLICATIONS

Big Data Testing or Hadoop Testing can be broadly divided into three steps.

Step 1 : Data Staging Validation

The first step in this big data testing tutorial is referred as pre-Hadoop stage involves process validation.

- Data from various source like RDBMS, weblogs, social media, etc. should be validated to make sure that correct data is pulled into the system
- Comparing source data with the data pushed into the Hadoop system to make sure they match
- Verify the right data is extracted and loaded into the correct HDFS location

Step 2 : “MapReduce” Validation

The second step is a validation of "MapReduce". In this stage, the Big Data tester verifies the business logic validation on every node and then validating them after running against multiple nodes, ensuring that the

- Map Reduce process works correctly
- Data aggregation or segregation rules are implemented on the data
- Key value pairs are generated
- Validating the data after the Map-Reduce process

Step 3 : Output Validation Phase

The final or third stage of Hadoop testing is the output validation process. The output data files are generated and ready to be moved to an EDW (Enterprise Data Warehouse) or any other system based on the requirement.

Activities in the third stage include

- To check the transformation rules are correctly applied
- To check the data integrity and successful data load into the target system
- To check that there is no data corruption by comparing the target data with the HDFS file system data

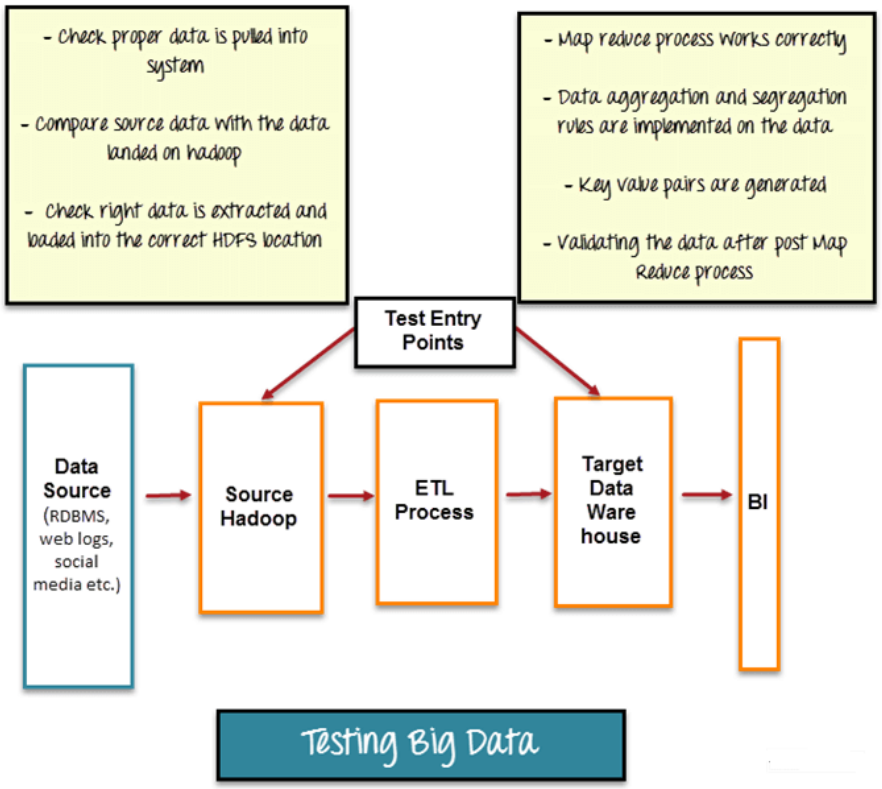



Fig. 5.1 BIG DATA TESTING

## 5.4 BENEFITS OF BIG DATA TESTING STRATEGY

- **Data Validation Testing:** Every organization strives for accurate data for business planning, forecasting and decision-making. This data needs to be validated for its correctness in any big data application. This validation process should confirm that:

  - the data injection process is error-free
  - complete and correct data is loaded to the big data framework
  - the data process validation is working properly based on the designed logic
  - the data output in the data access tools is accurate as per the requirement

- **Improved Business Decisions:** Accurate data is the pillar for crucial business decisions. When the right data goes in the hands of genuine people, it becomes a positive feature. It helps in analysing all kinds of risks and only the data that contribute to the decision-making process comes into the picture, and ultimately becomes a great aid to make sound decisions.

- **Cost-Effective Storage:** Behind every big data application, there are multiple machines that are used to store the data injected from different servers into the big data

framework. Every data requires storage-and storage doesn't come cheap. That's why it's important to thoroughly validate if the injected data is properly stored in different nodes based on the configuration, such as data replication factor and data block size. Keep in mind that any data that is not well structured or in bad shape requires more storage. Once that data is tested and is structured, the less storage it consumes, thus ultimately becoming more cost-effective.

- **Right Data at the Right Time:** Big data framework consists of multiple components. Any component can lead to bad performance in data loading or processing. No matter how accurate the data may be, it is of no use if it is not available at the right time. Applications that undergo load testing with different volumes and varieties of data can quickly process a large amount of data and make the information available when required.

- **Reduces Deficit and Boosts Profits:** Indigent big data becomes a major loophole for the business as it is difficult to determine the cause and location of errors. On the other hand, accurate data improves the overall business, including the decision-making process. Testing such data isolates the useful data from the unstructured or bad data, which will enhance customer services and boost business revenue.

## 5.5 BIG DATA TESTING CHALLENGES

Challenges faced when testing unstructured data are expected, especially when new to implementing tools used in big data scenarios.

- **Heterogeneity and Incompleteness of Data**

  - Problem: Many businesses today are storing exabytes of data in order to conduct daily business. Testers must audit this voluminous data to confirm its accuracy and relevance for the business. Manual testing this level of data, even with hundreds of QA testers, is impossible.
  - Solution: Automation in big data is essential to your big data testing strategy. In fact, data automation tools are designed to review the validity of this volume of data. Make sure to assign QA engineers skilled in creating and executing automated tests for big data applications.

- **High Scalability**

  - Problem: A significant increase in workload volume can drastically impact database accessibility, processing and networking for the big data application. Even though big data applications are designed to handle enormous amounts of data, it may not be able to handle immense workload demands.
  - Solution: Your data testing methods should include the following testing approaches:

  - Clustering Techniques: Distribute large amounts of data equally among all nodes of a cluster. These large data files can then be easily split into different chunks and stored in

different nodes of a cluster. By replicating file chunks and storing within different nodes, machine dependency is reduced.

- Data Partitioning: This automation in big data approach is less complex and is easier to execute. Your QA testers can conduct parallelism at the CPU level through data partitioning.

- **Test Data Management**

  - Problem: It is not easy to manage test data when it's not understood by your QA testers. Tools used in big data scenarios can only carry your team so far when it comes to migrating, processing and storing test data-that is, if your QA team doesn't understand the components within the big data system.
  - Solution: First, your QA team should coordinate with both your marketing and development teams in order to understand data extraction from different resources and data filtering as well as pre and post-processing algorithms. Provide proper training to your QA engineers designated to run test cases through your big data automation tools so that test data is always properly managed.

# CHAPTER 6

# Project Workflow

## 6.1 Setup Linux Machine using Oracle VirtualBox and CentOS

- **Virtual Box**

  VirtualBox is a powerful x86 and AMD64/Intel64 virtualisation product for enterprise as well as home use. Not only is VirtualBox an extremely feature rich, high performance product for enterprise customers, it is also the only professional solution that is freely available as Open Source Software under the terms of the GNU General Public License (GPL) version 2. Presently, VirtualBox runs on Windows, Linux, Macintosh, and Solaris hosts and supports a large number of guest operating systems including but not limited to Windows (NT 4.0, 2000, XP, Server 2003, Vista, Windows 7, Windows 8, Windows 10), DOS/Windows 3.x, Linux (2.4, 2.6, 3.x and 4.x), Solaris and OpenSolaris, OS/2, and OpenBSD.

  VirtualBox is being actively developed with frequent releases and has an ever-growing list of features, supported guest operating systems and platforms it runs on. VirtualBox is a community effort backed by a dedicated company: everyone is encouraged to contribute while Oracle ensures the product always meets professional quality criteria.


Figure 6.1.1 Oracle VM VirtualBox

- **CentOS**

The CentOS Project invites you to be a part of the community as a contributor. There are many ways to contribute to the project, from documentation, QA, and testing to coding changes for SIGs, providing mirroring or hosting, and helping other users.

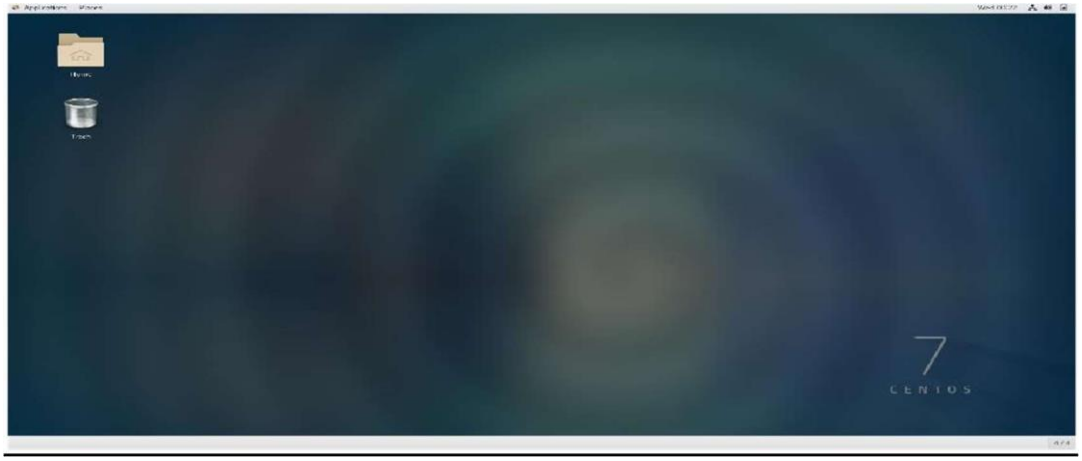



Figure 6.1.2 CentOS

- **Installation:**

Download the appropriate version of both the packages. The 64-bit architecture has been used for this demonstration, so download the software accordingly.
Install VirtualBox and open it. Click on the **new** to set up the new VM.

The ISO image of the downloaded CentOS has to be linked to the newly created virtual machine.

Select "**Install CentOS Linux 7**" and proceed.

After finishing the initial setup, you need to execute some additional steps. First, you need to accept the OS EULA.

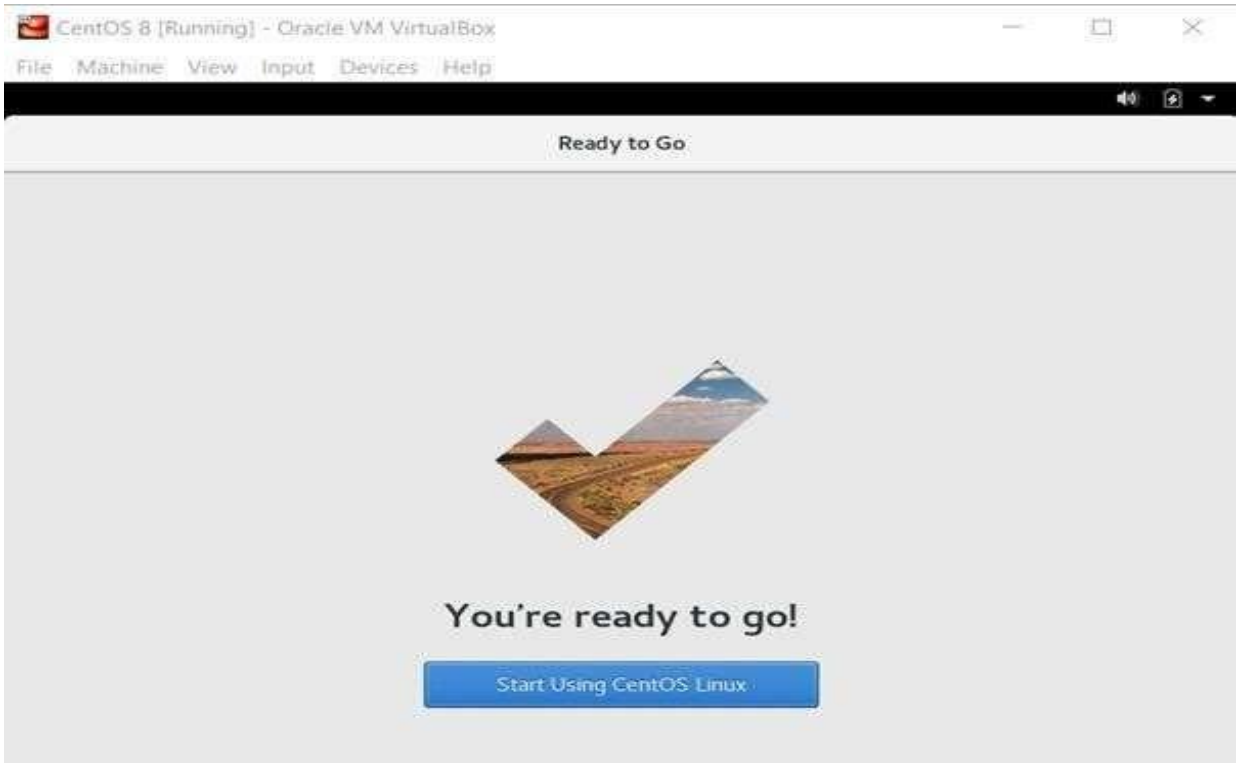



Figure 6.1.3 Installed CentOS

## 6.2 Hadoop Installation and Single Node Cluster Setup

- **Hadoop**

  Hadoop is an open-source framework that allows to store and process big data in a distributed environment 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.

- **Hadoop Installation**


Figure 6.2.1 Hadoop Installation

Step 1: Download the Java 8 Package. Save this file in your home directory.

Step 2: Extract the Java Tar File.

***Command*:** tar -xvf jdk-8u101-linux-i586.tar.gz

**Step 3: Download the Hadoop 2.7.3 Package.**

***Command*:** wgethttps://archive.apache.org/dist/hadoop/core/hadoop-2.7.3/hadoop-2.7.3.tar.gz

Step 4: Extract the Hadoop tar File.

***Command***: tar -xvf hadoop-2.7.3.tar.gz

Step 5: Add the Hadoop and Java paths in the bash file (.bashrc).

Open**. bashrc** file. Now, add Hadoop and Java Path as shown below.

***Command*:** vi .bashrc

Then, save the bash file and close it.

For applying all these changes to the current Terminal, execute the source command.

***Command*:** source .bashrc

To make sure that Java and Hadoop have been properly installed on your system and can be accessed through the Terminal, execute the java -version and hadoop version commands.

***Command*:** java –version

***Command*:** hadoop version

Step 6: Edit the Hadoop Configuration files.

***Command:*** cd hadoop-2.7.3/etc/hadoop/

***Command:*** ls

All the Hadoop configuration files are in **hadoop-2.7.3/etc/hadoop** directory.

Step 7: Open *core-site.xml* and edit the property mentioned below inside configuration tag:

*core-site.xml* informs Hadoop daemon where NameNode runs in the cluster. It contains configuration settings of Hadoop core such as I/O settings that are common to HDFS & MapReduce.

***Command*:** vi core-site.xml

Step 8: Edit *hdfs-site.xml* and edit the property mentioned below inside configuration tag:

*hdfs-site.xml* contains configuration settings of HDFS daemons (i.e. NameNode, DataNode, Secondary NameNode). It also includes the replication factor and block size of HDFS.

***Command*:** vi hdfs-site.xml

Step 9: Edit the *mapred-site.xml* file and edit the property mentioned below inside configuration tag:

*mapred-site.xml* contains configuration settings of MapReduce application like number of JVM that can run in parallel, the size of the mapper and the reducer process, CPU cores available for a process, etc.

In some cases, mapred-site.xml file is not available. So, we have to create the mapred-site.xml file using mapred-site.xml template.

***Command*:** cp mapred-site.xml.template mapred-site.xml

***Command*:** vi mapred-site.xml.

Step 10: Edit *yarn-site.xml* and edit the property mentioned below inside configuration tag:

*yarn-site.xml* contains configuration settings of ResourceManager and NodeManager like application memory management size, the operation needed on program & algorithm, etc.

***Command*:** vi yarn-site.xml

Step 11: Edit *hadoop-env.sh* and add the Java Path as mentioned below:

*hadoop-env.sh* contains the environment variables that are used in the script to run Hadoop like Java home path, etc.

***Command*:** vi hadoop–env.sh

Step 12: Go to Hadoop home directory and format the NameNode.

***Command*:** cd

***Command*:** cd hadoop-2.7.3

***Command*:** bin/hadoop namenode -format

This formats the HDFS via NameNode. This command is only executed for the first time. Formatting the file system means initializing the directory specified by the dfs.name.dir variable.

Never format, up and running Hadoop filesystem. You will lose all your data stored in the HDFS.

Step 13: Once the NameNode is formatted, go to hadoop-2.7.3/sbin directory and start all the daemons.

***Command:*** cd hadoop-2.7.3/sbin

Either you can start all daemons with a single command or do it individually.

***Command:*** ./start-all.sh

The above command is a combination of ***start-dfs.sh, start-yarn.sh*** & ***mr-jobhistory-daemon.sh***

Or you can run all the services individually as below:

Start NameNode:

The NameNode is the centerpiece of an HDFS file system. It keeps the directory tree of all files stored in the HDFS and tracks all the file stored across the cluster.

***Command:*** ./hadoop-daemon.sh start namenode

Start DataNode:

On startup, a DataNode connects to the Namenode and it responds to the requests from the Namenode for different operations.

***Command:*** ./hadoop-daemon.sh start datanode

Start ResourceManager:

ResourceManager is the master that arbitrates all the available cluster resources and thus helps in managing the distributed applications running on the YARN system. Its work is to manage each NodeManagers and the each application's ApplicationMaster.

***Command:*** ./yarn-daemon.sh start resourcemanager

Start NodeManager:

The NodeManager in each machine framework is the agent which is responsible for managing containers, monitoring their resource usage and reporting the same to the ResourceManager.

***Command:*** ./yarn-daemon.sh start nodemanager

Start JobHistoryServer:

JobHistoryServer is responsible for servicing all job history related requests from client.

***Command*:** ./mr-jobhistory-daemon.sh start historyserver

Step 14: To check that all the Hadoop services are up and running, run the below command.

***Command:*** jps

Step 15: Now open the Mozilla browser and go to localhost:50070/dfshealth.html to check the NameNode interface.

- **Hadoop Architecture**

Hadoop has a Master-Slave Architecture for data storage and distributed data processing using MapReduce and HDFS methods. The Hadoop architecture is a package of the file system, MapReduce engine and the HDFS (Hadoop Distributed File System). The MapReduce engine can be MapReduce/MR1 or YARN/MR2. A Hadoop cluster consists of a single master and multiple slave nodes. The master node includes Job Tracker, Task Tracker, NameNode, and DataNode whereas the slave node includes DataNode and TaskTracker.

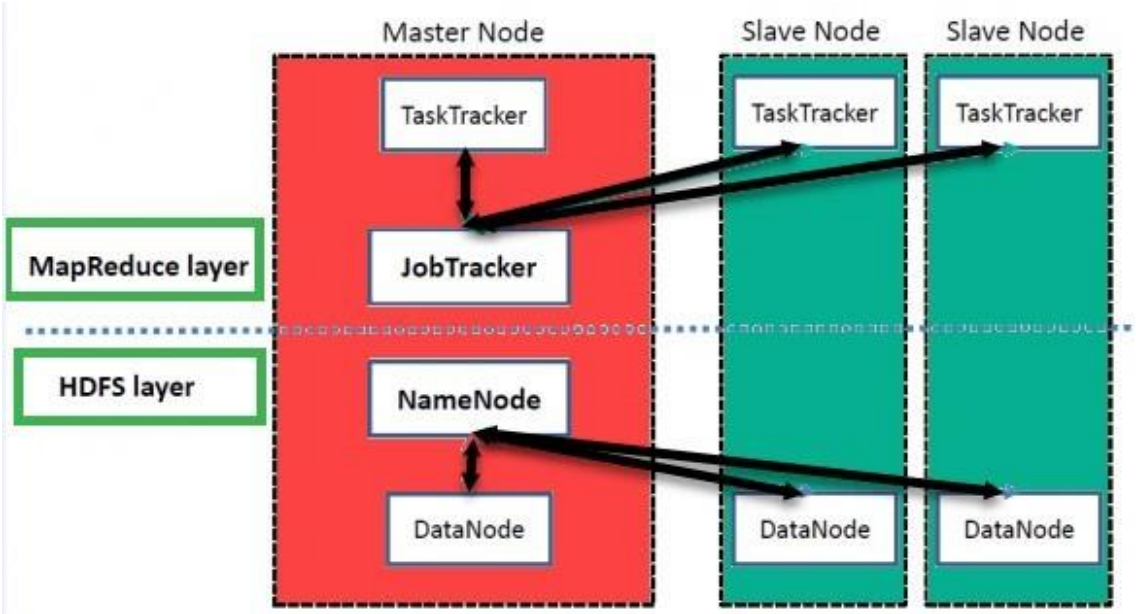



Figure 6.2.2 Hadoop Architecture

- **Single Node Cluster**

  Single node cluster means only one DataNode running and setting up all the NameNode, DataNode, ResourceManager and NodeManager on a single machine.
  It can easily and efficiently test the sequential workflow in a smaller environment as compared to large environments which contains terabytes of data distributed across hundreds of machines.

  Hadoop cluster consists of three components –

- Master Node – Master node in a hadoop cluster is responsible for storing data in HDFS and executing parallel computation the stored data using MapReduce. Master Node has 3 nodes – NameNode, Secondary NameNode and JobTracker. JobTracker monitors the parallel processing of data using MapReduce while the NameNode handles the data storage function with HDFS. NameNode keeps a track of all the information on files (i.e. the metadata on files) such as the access time of the file, which user is accessing a file on current time and which file is saved in which hadoop cluster. The secondary NameNode keeps a backup of the NameNode data.

- Slave/Worker Node- This component in a hadoop cluster is responsible for storing the data and performing computations. Every slave/worker node runs both a TaskTracker and a DataNode service to communicate with the Master node in the cluster. The DataNode service is secondary to the NameNode and the TaskTracker service is secondary to the JobTracker.

- Client Nodes – Client node has hadoop installed with all the required cluster configuration settings and is responsible for loading all the data into the hadoop cluster. Client node submits mapreduce jobs describing on how data needs to be processed and then the output is retrieved by the client node once the job processing is completed.

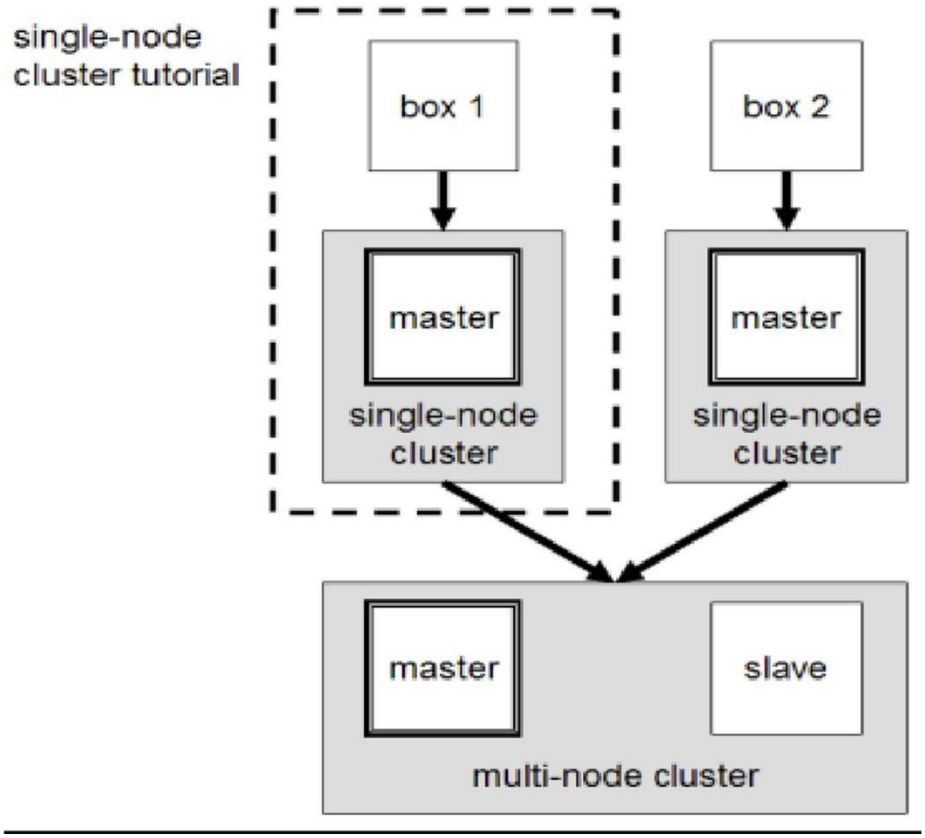
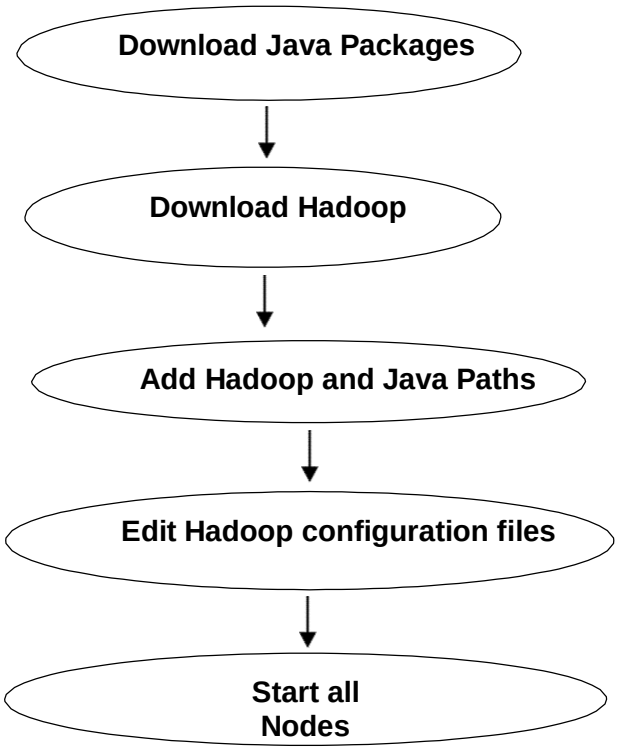



Figure 6.2.3 Single Node Cluster

- **Single Node Cluster Setup**



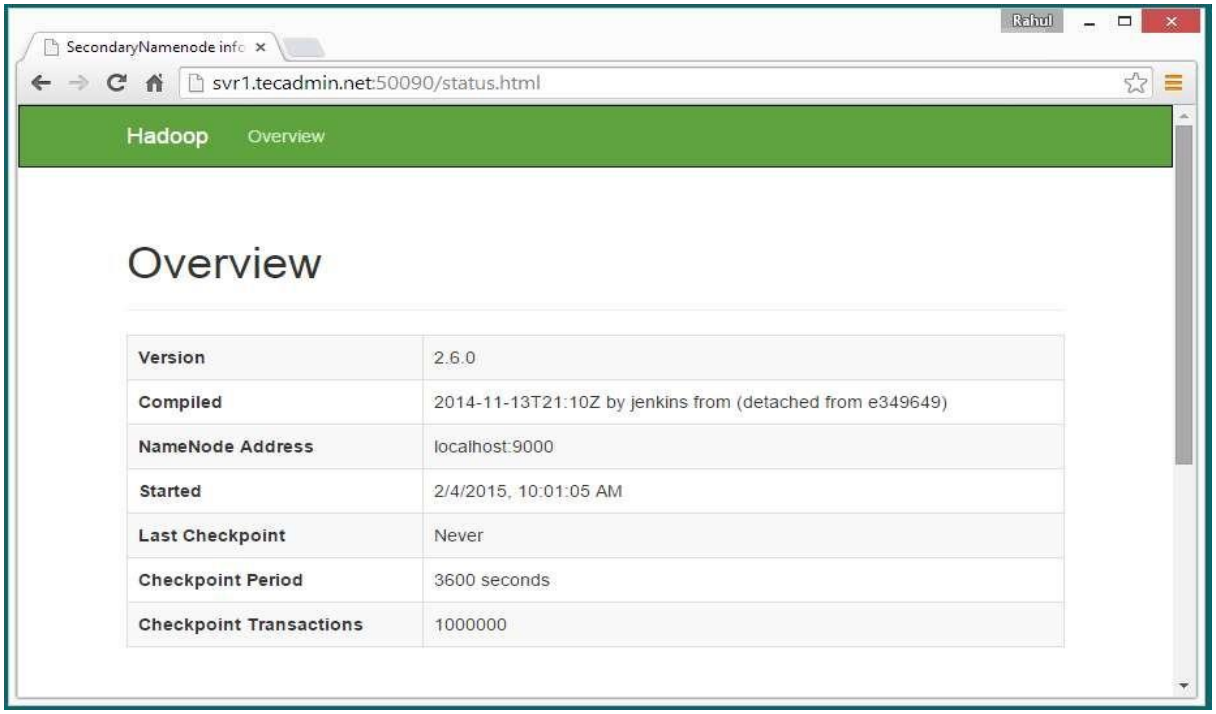



Figure 6.2.4 Single Node Cluster Setup

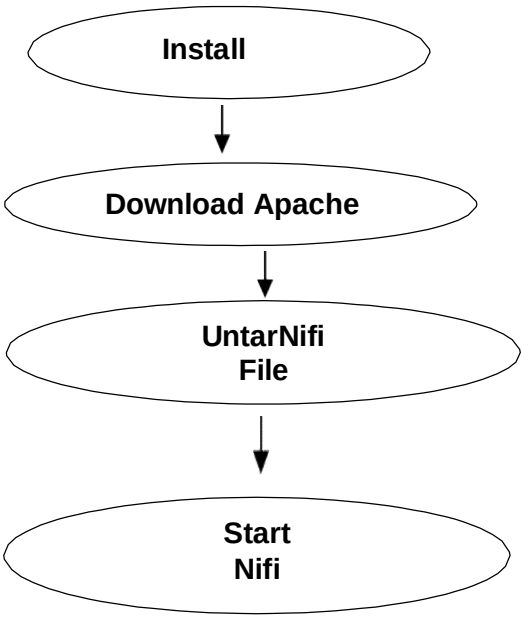
## 6.3 Nifi Installation

- **Apache Nifi**

  Apache NiFi is an open source data ingestion platform. It was developed by NSA and is now being maintained and further development is supported by Apache foundation. It is based on Java, and runs in Jetty server. It is licensed under the Apache license version 2.0.The guide we are giving in this tutorial is intended to provide knowledge how to work with NiFi. To work with NiFi, you should have the basic knowledge of Java, Data ingestion, transformation, and ETL. You should also be familiar with the regex pattern, web server, and platform configuration.


Figure 6.3.1 Apache Nifi

- **Nifi Setup**



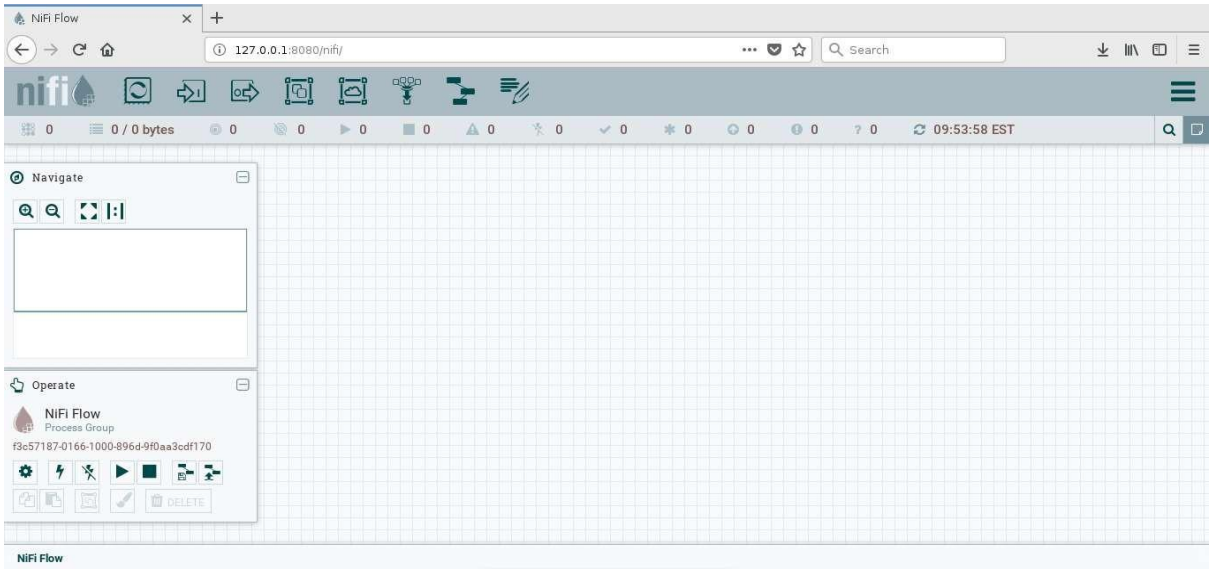



Figure 6.3.2 Nifi Setup

## 6.4 Hive Installation

- **Apache Hive**

  Apache Hive is a data ware house system for Hadoop that runs SQL like queries called HQL (Hive query language) which gets internally converted to map reduce jobs. Hive was developed by Facebook. It supports Data definition Language, Data Manipulation Language and user defined functions. Hive abstracts the complexity of Hadoop MapReduce. Basically, it provides a mechanism to project structure onto the data and perform queries written in HQL (Hive Query Language) that are similar to SQL statements. Internally, these queries or HQL gets converted to map reduce jobs by the Hive compiler.


Figure 6.4.1 Apache Hive

- **Hive Architecture**

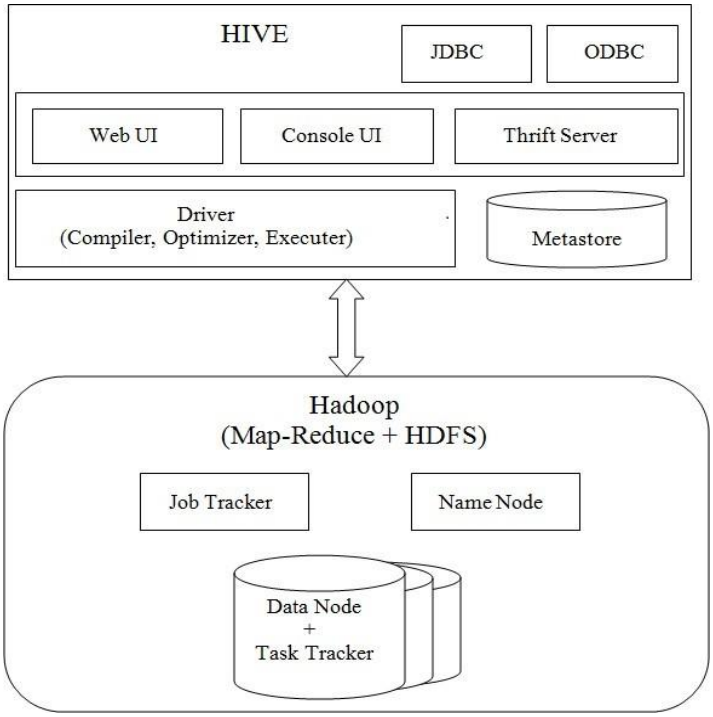



Figure 6.4.2 Hive Architecture

- **Hive Setup**

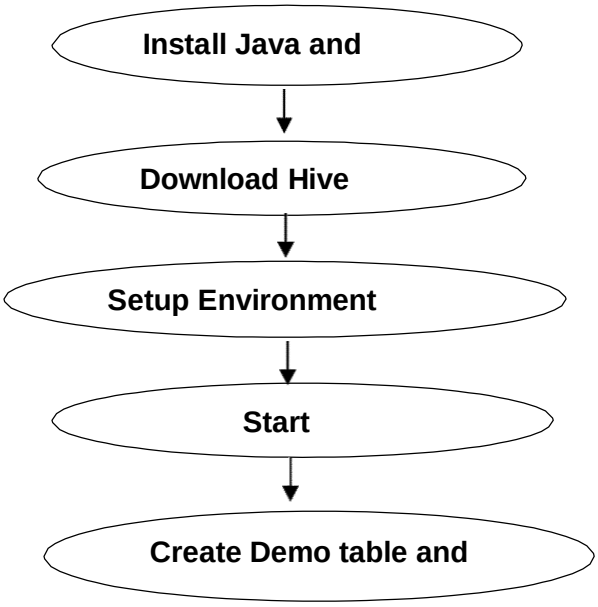



Figure 6.4.3 Hive Setup

## 6.5 Analysis of Data

The analysis involves following steps.

- Rating of action movies on user gender.
- Rating of adventure movies on user gender.
- Highest rated movie in a year.
- Maximum FB likes in a year.

For analyzing dataset, we have created the views of each table. The purpose of creating the view is to minimize the execution time. For example if there is a table with 1000 columns and we want to access 10 columns from that. In this case a view of those 10 columns is created. Instead of scanning the entire table for 10 columns, query can be executed on the view to minimize the execution time. In our analysis these views are then joined on the condition to get the result.

The figure 6.5.1 shows the result of average rating given by the female gender for each action category movie. We have used three views: Movie_Action which has the columns movie_id, title, genre as Action from Genre dataset. Female_view has the colums user_id and gender as F from user dataset. Rat_F has movie_id, ratings, gender.

| | | | | |
|---|---|---|---|---|
| 3654 | Guns of Navarone The (1961) | Action\|Drama\|War | 4.061224489795919 | F |
| 1221 | Godfather: Part II The (1974) | Action\|Crime\|Drama | 4.04093567251462 | F |
| 2194 | Untouchables The (1987) | Action\|Crime\|Drama | 4.021164021164021 | F |
| 110 | Braveheart (1995) | Action\|Drama\|War | 4.016483516483516 | F |
| 1910 | I Went Down (1997) | Action\|Comedy\|Crime | 4.0 | F |
| 139 | Target (1995) | Action\|Drama | 4.0 | F |
| 3137 | Sea Wolves The (1980) | Action\|War | 4.0 | F |
| 2924 | Drunken Master (Zui quan) (1979) | Action\|Comedy | 4.0 | F |
| 251 | Hunted The (1995) | Action | 4.0 | F |
| 2823 | Spiders The (Die Spinnen 1. Teil: Der Goldene See) (1919) | Action\|Drama | 4.0 | F |
| 2756 | Wanted: Dead or Alive (1987) | Action | 4.0 | F |
| 2737 | Assassination (1987) | Action | 4.0 | F |
| 390 | Faster Pussycat! Kill! Kill! (1965) | Action\|Comedy\|Drama | 4.0 | F |
| 1832 | Heaven's Burning (1997) | Action\|Drama | 4.0 | F |
| 1434 | Stranger The (1994) | Action | 4.0 | F |
| 624 | Condition Red (1995) | Action\|Drama\|Thriller | 4.0 | F |
| 1287 | Ben-Hur (1959) | Action\|Adventure\|Drama | 3.9765625 | F |
| 1277 | Cyrano de Bergerac (1990) | Action\|Drama\|Romance | 3.948905109489051 | F |
| 1387 | Jaws (1975) | Action\|Horror | 3.946875 | F |

Figure 6.5.1 Average Rating By Female Gender

The figure 6.5.2 shows the result of which movie has got maximum Facebook likes in a particular year. The views created here are Movie_FB with year, title and fb_likes from Movie dataset. Movie_FB_Max with year and maximum fb_likes for that year.

| | | |
|---|---|---|
| 1967 | 3000 | Point Blank |
| 1968 | 24000 | 2001: A Space Odyssey |
| 1969 | 548 | Sweet Charity |
| 1970 | 690 | Waterloo |
| 1971 | 819 | Escape from the Planet of the Apes |
| 1972 | 43000 | The Godfather |
| 1973 | 18000 | The Exorcist |
| 1974 | 14000 | Young Frankenstein |
| 1974 | 14000 | The Godfather: Part II |
| 1975 | 32000 | One Flew Over the Cuckoo's Nest |
| 1976 | 35000 | Taxi Driver |
| 1977 | 33000 | Star Wars: Episode IV - A New Hope |
| 1978 | 13000 | Grease |
| 1979 | 23000 | Alien |
| 1980 | 37000 | The Shining |
| 1981 | 16000 | Raiders of the Lost Ark |
| 1982 | 34000 | Blade Runner |
| 1982 | 34000 | E.T. the Extra-Terrestrial |
| 1983 | 19000 | Scarface |

Figure 6.5.2 Maximum Facebook Like

## 6.6 Visualisation

- **Tableau**

  Tableau is a powerful and fastest growing data visualization tool used in the Business Intelligence Industry. It helps in simplifying raw data in a very easily understandable format. Tableau helps create the data that can be understood by professionals at any level in an organization. It also allows non-technical users to create customized dashboards. Data analysis is very fast with Tableau tool and the visualizations created are in the form of dashboards and worksheets.

Figure 6.6.1 Tableau

- **Visualisation Outputs**

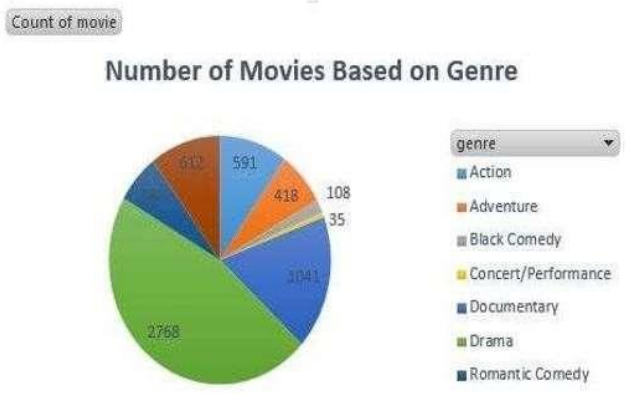



Figure 6.6.2 Count of movie


Figure 6.6.3 Count of movie

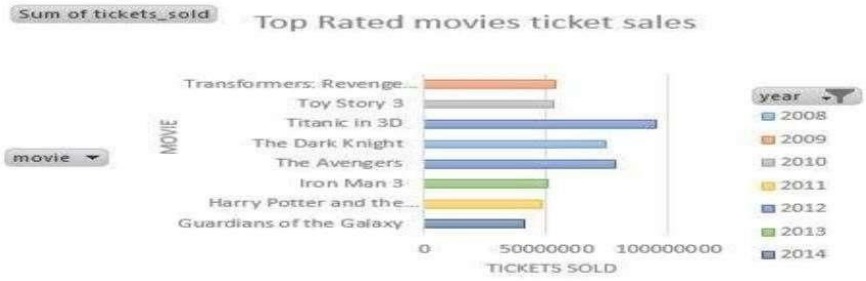



Figure 6.6.4 Top Rated movies ticket sales

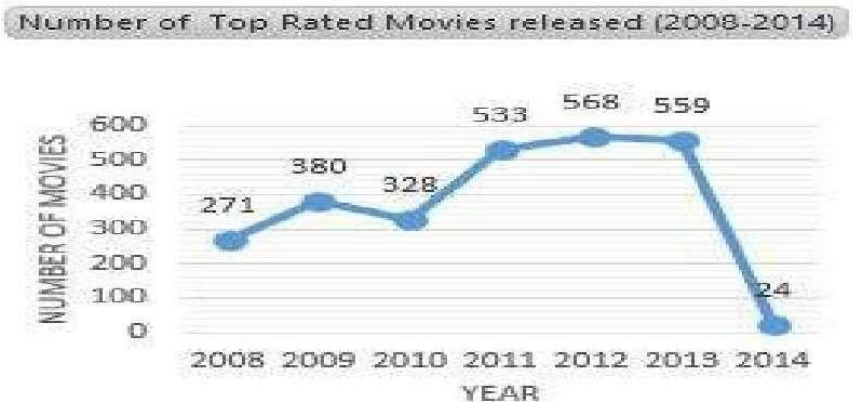



Figure 6.6.5 Top Rated movies

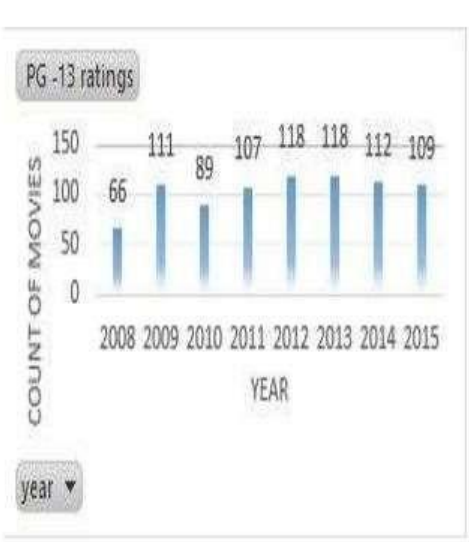



Figure 6.6.6 Count of Movies

# CHAPTER 7

# CONCLUSION AND FUTURE WORK

Hive is mainly used to process large amount of data. It is faster than SQL on low-cost machine. Hive performance is poor on smaller dataset but as the data size increases its processing time decreases. SQL is efficient and more robust for smaller data.

In order to find the relationship between year, movie title, imdb rating facebook likes, genre the dataset is analysed using HiveQL and its performance is compared with SQL performance. The result states that Hive performs better than SQL for larger dataset. For the future work we can consider the director, actor to predict which director has directed the best rated movie and which actor has got role in that. User ratings can be determined for other category.

# REFRENCES


1. Lucio Grandinetti, Seyedeh Leili Mirtaheri, Reza Shahbazian (Eds.), *High-Performance Computing and Big Data Analysis*, Second International Congress, TopHPC 2019 Tehran, Iran, April 23–25, 2019 Revised Selected Papers, Springer.
2. Bandhakavi, A., Wiratunga, N., & Massie, S. (2018). Emotion-aware polarity lexicons for Twitter sentiment analysis. *Expert Systems*, e12332.
3. Becken, S., Alaei, A. R., & Wang, Y. (2019). Benefits and pitfalls of using tweets to assess destination sentiment. *Journal of Hospitality and Tourism Technology*.
4. Zongben Xu, Xinbo Gao, Qiguang Miao, Yunquan Zhang, Jiajun Bu (Eds.)., *Big Data*, 6th CCF Conference, Big Data 2018 Xi'an, China, October 11–13, 2018 Proceedings,Springer.
5. Basiri, M. E., Nemati, S., Abdar, M., Asadi, S., & Acharrya, U. R. (2021). A novel fusion-based deep learning model for sentiment analysis of COVID-19 tweets. *Knowledge-Based Systems*, 107242.
6. T. A. Ashwitha, A. P. Rodrigues and N. N. Chiplunkar, "Movie Dataset Analysis Using Hadoop-Hive," *2017 2nd International Conference on Computational Systems and Information Technology for Sustainable Solution (CSITSS)*, 2017, pp. 1-5, doi: 10.1109/CSITSS.2017.8447828.
7. Kune, R., Konugurthi, P. K., Agarwal, A., Chillarige, R. R., & Buyya, R. (2016). The anatomy of big data computing. *Software: Practice and Experience*, *46*(1), 79-105.
8. Wang, J., Xu, C., Zhang, J., & Zhong, R. (2021). Big data analytics for intelligent manufacturing systems: A review. *Journal of Manufacturing Systems*.
9. Costa, R. L. D. C., Moreira, J., Pintor, P., dos Santos, V., & Lifschitz, S. (2021). A Survey on Data-driven Performance Tuning for Big Data Analytics Platforms. *Big Data Research*, *25*, 100206.
10. Uzunkaya, C., Ensari, T., & Kavurucu, Y. (2015). Hadoop ecosystem and its analysis on tweets. *Procedia-Social and Behavioral Sciences*, *195*, 1890-1897.
11. Mohammed M. Alani, Hissam Tawfik, Mohammed Saeed, Obinna Anya, *Applications of Big Data Analytics*, Trends, Issues, and Challenges, Springer.

12. Zhai, Y., Tchaye-Kondi, J., Lin, K. J., Zhu, L., Tao, W., Du, X., & Guizani, M. (2021). Hadoop Perfect File: A fast and memory-efficient metadata access archive file to face small files problem in HDFS. *Journal of Parallel and Distributed Computing*.
13. FEI HU, *Big Data Sharing Storage and Security*, CRC Press Taylor & Francis Group.
14. Kalia, K., & Gupta, N. (2020). Analysis of hadoop MapReduce scheduling in heterogeneous environment. *Ain Shams Engineering Journal*.
15. Ashwitha T AAnisha P Rodrigues Niranjan N Chiplunkar, *Movie Dataset Analysis using Hadoop-Hive*.
16. Lucas Filho, E. R., de Almeida, E. C., Scherzinger, S., & Herodotou, H. (2021). Investigating Automatic Parameter Tuning for SQL-on-Hadoop Systems. *Big Data Research*, *25*, 100204.
17. Ammar Fuad, Alva Erwin, Henru Purnomo Ipung, *Processing Performance on Apache Pig, Apache Hive and MySQL Cluster*, IEEE, 2014 International Conference on Information, Communication Technology and System.
18. Shujia Zhou et al., *Visualization and Diagnosis of Earth Science Data through Hadoop and Spark,* 978- 1-4673-9005-7/16/2016 IEEE International Conference on Big Data.
19. Karan Sachdeva et al., *Comparison of Data Processing Tools in Hadoop*, IEEE, 2016 International Conference on Electrical, Electronics,Communication, Computer and Optimization Techniques.
20. Aditya Bhardwaj et al., *Big Data Emerging Technologies: A CAseStudy with Analyzing Twitter Data using Apache Hive*, IEEE, 2015 RAECS UIET Panjab University Chandigarh