

BIG DATA FOUNDATION

A REPORT

submitted by

B.SAIKALYAN (18MIS1113)

in partial fulfilment for the award

of

M. Tech. Software Engineering (Integrated)

School of Computer Science and Engineering



VIT[®]
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

MAY 2022



VIT[®]
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

School of Computer Science and Engineering

DECLARATION

I hereby declare that the project entitled “**BIG DATA FOUNDATION(BDA) BY NASSCOM**” submitted by me to the School of Computer Science and Engineering, Vellore Institute of Technology, Chennai Campus, Chennai 600127 in partial fulfilment of the requirements for the award of the degree of **Master of Technology - Software Engineering (Integrated)** is a record of bonafide work carried out by me. I further declare that the work reported in this report has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma of this institute or of any other institute or university.

Signature

B.SAIKALYAN (18MIS1113)



VIT[®]
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

School of Computer Science and Engineering

CERTIFICATE


The project report entitled “**BIG DATA FOUNDATION(BDA) BY NASSCOM**” is prepared and submitted by **B.SAIKALYAN (Register No: 18MIS1113)**. It has been found satisfactory in terms of scope, quality and presentation as partial fulfilment of the requirements for the award of the degree of **Master of Technology – Software Engineering (Integrated)** in Vellore Institute of Technology, Chennai, India.


Examined by:


Examiner I


Examiner II

INSERT THE CERTIFICATE OF MERIT OBTAINED FROM THE INDUSTRY WHERE YOU DID YOUR INTERNSHIP









CERTIFICATE


This is to certify that

SAIKALYAN BANDEPALLI 18MIS1113

has successfully cleared the assessment on

Big Data (BDA) Foundation


Aligned to Competency Standards developed by IT-ITeS Sector Skills Council
NASSCOM in collaboration with Industry and approved by Government.



FSP/2021/8/1503658

15/08/2021

Date of issue



CEO, IT-ITeS Sector Skills Council
NASSCOM

Gold Category: 70% and above score.

Detailed Scorecard included







Certificate Details

Candidate Name	Saikalyan Bandepalli 18Mis1113
Assessment/Course Name	Big Data (BDA) Foundation
Date of Issue	15/08/2021
Certification ID	FSP/2021/8/1503658
Category Gold >=70% / Silver 60%-69% / Bronze 50%-59%	Gold



FSP/2021/8/1503658

Assessment Score

Module Name/NOS ID	NSQF Level	Maximum Marks	Marks Obtained	Percentage
M001	NA	12.00	10.00	83.33
M002	NA	35.00	32.00	91.43
M003	NA	53.00	35.00	66.04
Total		100.00	77.00	77.00



ACKNOWLEDGEMENT

I would like to thank Dr. Asnath Vicky Phamila Y, Head of the Department (HoD), M.Tech Software Engineering (5 year integrated), SCSE, VIT Chennai, Dr. Ganesan R, Dean of the School of Computer Science & Engineering, VIT Chennai, Dr. Geetha S, Associate Dean of the School of Computer Science & Engineering, VIT Chennai for giving me this opportunity to learn Big Data Foundation. It was a great experience to learn this course where I learned how to analyse the real time big data.

CONTENTS

Chapter	Title	Page
	Title Page	1
	Declaration	2
	Certificate	3
	Industry certificate	4
	Acknowledgement	5
	Table of contents	6
	List of Figures	8
	List of Abbreviation	10
1	Abstract	11
2	Introduction	12
3	Introduction to Big Data Analytics	13
4	Big Data Fundamentals and Platforms	17
5	Big Data Processing, Management and Analytics	23
6	Conclusion	27
7	References	28
8	Appendix – I	29

LIST OF FIGURES

Title	Page
Fig 1: how much quantity have been purchased over the sales cost more than 500.	03
Fig 2: how much total quantity have been purchased by Sub-Category wise.	03
Fig 3: how much total quantity have been purchased by Sub-Category wise output.	03
Fig 4: Implemented K-means Clustering	04
Fig 5. Inference Office, supplies, technologies in pie chart.	04
Fig 6. Inference on Total number of Items sold in region wise	04
Fig 7. Decision Tree classifier output	04
Fig 8. Linear Regression Sales Vs Profit	06
Fig 9. Hive output 1	06
Fig 10. Hive Output 2	06
Fig 11. Visualization on Tableau “Sales and Profit by Ship Mode”	06
Fig 12. Visualization on Tableau “Sales by Category and sub- category”	07
Fig 13. Visualization on Tableau “Profit, Quantity and discount by Sub-Category in Ship Mode”	08

LIST OF ABBREVIATIONS

Abbreviation	Expansion
HDFS	Hadoop Distributed File System
UDF	User Defined Function
DailyLogRetSPX	Daily Log Return of Standard & Poor's index

ABSTRACT

This internship program was an online internship program and we are assigned to complete the course named 'Big Data Foundation Course' offered by SSC NASSCOM and Digital Vidhya'. The task here is to complete the offered course to develop a fundamental knowledge on the vast field of big data. The Big Data Foundation course will establish an understanding of Big Data Analytics, Visualization, Data Processing and Management along with the fundamentals of different Big Data Platforms and provides some practical exposure to some of the popular platform like Hadoop and HDFS. Big Data Analytics (BDA) greatly helps companies in making informed decisions by analyzing huge datasets and uncovering hidden patterns, cryptic correlations, customer preferences, and market trends. Through this we can able to identify Big Data and its Business implications, list the components of Hadoop and Hadoop Eco-system, access and process data on distributed file system, manage job execution in Hadoop Environment, use the hive database to perform queries on the datasets, use mongoDB to retrieving processes, and perform Map Reduce queries on the dataset for the faster retrieval of data. So through this course, we will be able to proceed further either as Big Data ETL Engineer or Big Data Application Engineer.

INTRODUCTION

Big data analytics means the process of harnessing these large data sets to reveal hidden patterns, market trends, customer preferences, etc. With the help of big data analytics, business owners are empowered to derive values from information and make optimal business decisions. The large volume of data – both structured and unstructured – that inundates a business on a daily basis is referred to as big data. But it's not the quantity of data that matters. It's what the organizations do with the data that matters. Big data can be analysed to uncover insights that lead to more informed decisions and strategic business moves. Big data analytics is the use of advanced analytic techniques to very large, diverse big data sets, which can contain structured, semi-structured, and unstructured data, as well as data from many sources and sizes ranging from terabytes to zetta bytes. Big Data is described as data sets that are too large or complex for traditional relational databases to acquire, manage, and handle in a timely manner. Big data has a lot of volume, a lot of velocity, and a lot of variation. Characteristics of big data include high volume, high velocity and high variety. Sources of data are becoming more complex than those for traditional data because they are being driven by artificial intelligence (AI), mobile devices, social media and the Internet of Things (IoT). For example, the different types of data originate from sensors, devices, video/audio, networks, log files, transactional applications, web and social media — much of it generated in real time and at a very large scale

Introduction to Big Data Analytics

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

Big data – 4Vs

1. Volume
2. Velocity
3. Variety
4. Veracity
5. Value

1st V: 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'.

2nd V: 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. 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'.

3rd V: 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.

4th V: 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.

5th V: Value

Uncover Hidden Patterns, unknown correlations, customer preferences and other various important information. The value uncovered helps organizations, industries to create new products, to explore new market. Help companies streamline operations, improve marketing, enhance customer engagement, improve customer service. Extend the value of a predictive model by subsequently uncovering a virtually unfathomable combination of additional variables.

Traditional Grid Computing:

- Distribute the processing
- Worker Nodes sharing the same storage system, acting as a bottleneck.

Putting it all together:

- Storing lot of data (Big Data) is inexpensive on commodity hardware's
- Reading or writing 100 GB from a single disk takes 20min and it would only use 1/16th of the machine's CPU resources.

Introduction to distributed computing:

Distributed Computing is an environment in which a group of independent and geographically dispersed computer systems take part to solve a complex problem, each by solving a part of solution and then combining the result from all computers.

Characteristics of distributed computing:

- a) Resource sharing
 - i) Resource, Hardware, Data
- b) Concurrency
 - i) Multi programming
 - ii) Multi processing
- c) Scalability
 - i) Scalable to multiple computers
- d) Fault tolerant
 - i) Hardware Redundancy
 - ii) Software Recovery
 - iii) Data Recovery
- e) Transparency
 - i) Transparency ,location,concurrency,replication,failure,migration,performance,scaling

Industry Use cases of big data analytics:

Use Case:

Fraud detection.

Data:

Transaction data, 360 degree feedback.

Value:

Credit card companies or Banks already manage huge volume of data from individual Social Security number, income, account balances employment details, credit history and transaction history All this put together helps credit card companies to fight fraud in real-time Big Data architecture provides that scalability to analyze the incoming transaction against individual history and approve/decline the transaction and alert the account owner.

Use Case:

Call Centre Analysis

Data:

Call log, Transactional data

Value:

For decades, companies have been analysing call centre data for staffing agent performance, network management but with big data age, many new interesting software are being implemented today in attempt to take unstructured voice recordings and analyse them for content and sentiment Banks are applying text and sentiment analysis to this unstructured data, and looking for patters and trends Many banks are integrating this call centre data with their transactional data warehouse to reduce customer churn, and drive up sell cross sell customer monitoring alerts and fraud detection.

Key for implementing big data solution:

Define the problem → Define the impact → Decide on the infrastructure → Define the data requirements → Identify competence requirements → Decide on the implementation approach.

Reduntant physical infrastructure:

- Performance
- Availability
- Scalability
- Flexibility
- Cost

Design and Implement the Big data Solution:

- Data source and Access requirements
- Instrumentation strategy
- Real-time data access and analysis requirements
- Data management capabilities
- Data modeling capabilities
- Business Intelligence
- Advance analytic capabilities
- User experience requirement

Data Preparation:

- Provisioning an Analytical Workspace
- Acquiring, Cleansing, Aligning and analyzing the data.
- Transforming

Model Planning:

- Determine different Analytic Models
- Determine correlation and collinearity

Model Building:

- Massaging the datasets for testing, training and production
- Assessing the viability and reliability
- Developing, testing and refining the analytical models

Communicate Results:

- Ascertaining the quality and reliability
- Preparing reports

Operationalize:

- Delivering Reports
- Running pilot and analytical lab
- Implementing in production
- Integrating analytic scores

Big Data Fundamentals and Platforms

Data Ware House:

Data warehousing emphasizes the capture of data from different sources for access and analysis. Three main component of Data Warehouse. – Data sources from operational systems, such as Excel, ERP, CRP or financial applications – A data staging area where data is cleaned and ordered – A presentation area where data is warehoused.

Data may be: – Structured – Semi-structured –Unstructured data.

Types of Data Warehouse – Enterprise Data Warehouse: – Operational Data Store – Data Mart.

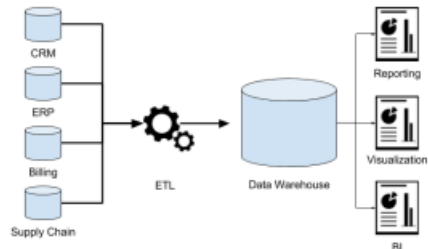


Fig 1: Typical Data Ware house

Step 1: Operational or transactional or day – to –day business data is gathered.

Step 2: This data is then integrated, cleaned up, transformed, and standardized through the process of Extraction, Transformation, and Loading (ETL).

Step 3: The transformed data is then loaded into enterprise data warehouse or data marts.

Step 4: A host of market leading business intelligence and analytics tools are then used to enable decision making from the use of ad-hoc queries, SQL, enterprise dashboards, data mining etc.

Terminologies Used in Big data:

In-memory analytics, In-Database processing, symmetric multiprocessor system, Massively parallel processing.

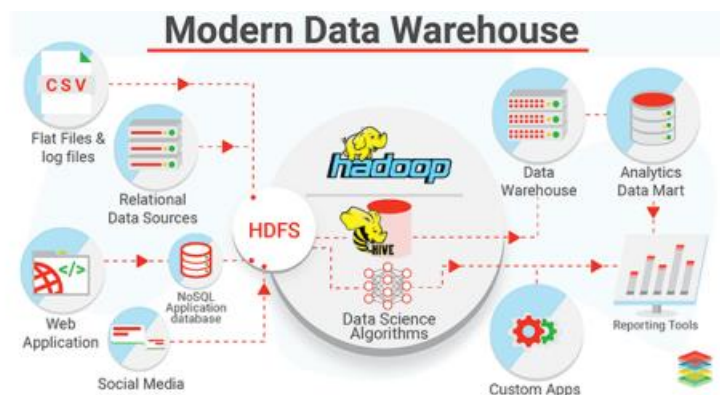


Fig 2: Modern Data Ware house

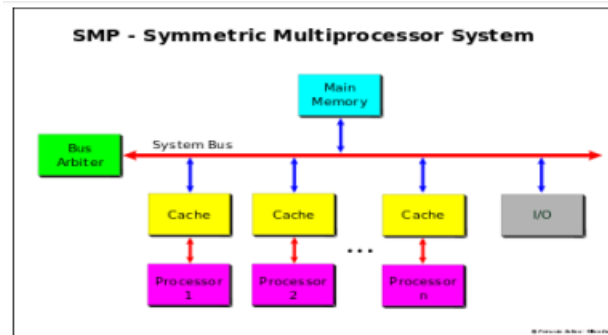


Fig 3: Parallel System

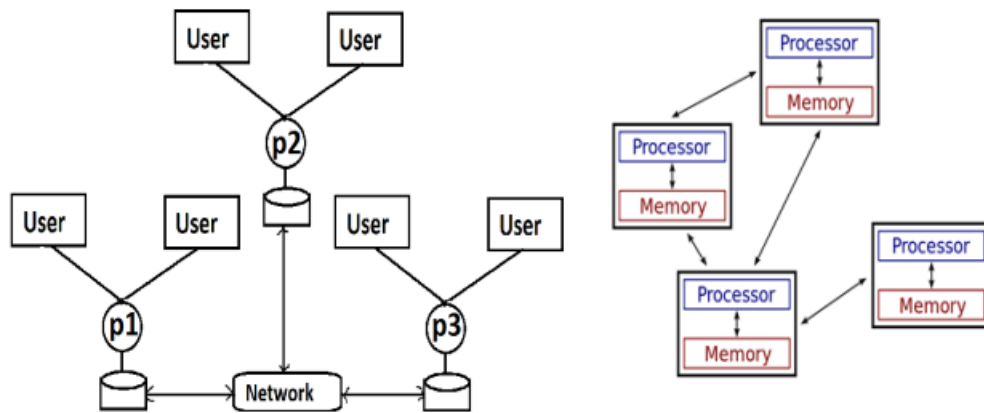


Fig 4: Distributed System

Hadoop:

Hadoop is an open-source software which is used for distributed computing. Hadoop is reliable and scalable. The framework of software library of Apache Hadoop makes the distributed computing easy of a very large dataset. It is computed across different clusters of the same computer using single programming models and sometimes different clusters of the different computer as well. This Hadoop will be able to scale up from a single computer server to multiple computer server which has its own local storage and computation. The library embedded is able

to detect the failures and handle them in the application layer instead of relying on the hardware for the high-availability delivery. Distributed computing takes places with the help of demons called “Name Node”, “Secondary Name Node”, “Data Node”, “Job Tracker”, “Task Tracker”.

Name Node has the block details i.e., which block is in which data node. It has the metadata of the dataset which is given for the distributed computing. It also has the edit log details and the file system image. Name Node is the master node. Data Node is the slave node. Secondary Name Node will periodically collect the data from the name node and helps when there is a name node failure. It doesn't do real time back up.

Task Tracker where the map task and reduce task takes place. It processes the program we write to compute the data which is given as the input in the form of key value pair. Task Tracker is the slave which reports to the Job tracker which is the master.

Hive:

There are many sub-projects in the Hadoop ecosystem which helps and one such is Hive. Hive is a platform which helps in development of the type scripts of SQL to perform the Map Reduce tasks. Hive which is also known as Hive Query Language to perform the Map Reduce task processes only the structured data which is of the form of rows and column. Hive is a tool with an infrastructure of data warehouse. It helps in summarizing the big data and helps in making the querying and analyzing the data very easy. Hive won't work for the processing of the real-time data for querying, it is not a relational database and won't work for the online transaction processing which is also known as OLTP. Hive helps in storing the schema in its database and after the data is processed it is then stored in HDFS. It is designed for the Online Analytical Processing. The syntax is similar to MySQL. Hive is very fast, extensible, familiar to MySQL and scalable.



Fig 5: Hive Architecture

Pig:

Pig raises the level of abstraction for processing large datasets. It is made up of two pieces:
 –The language used to express data flows, Called Pig Latin
 –The execution environment to run Pig Latin Programs. Pig Latin program is made up of a series of operation, or transformations, that are applied to the input data to produce output. Pig is scripting language for exploring large datasets. Pig was designed to be extensible.

Query: Load, For Each, Filter, Dump

SQOOP:

SQOOP is a command line tool which runs on bash or zsh. Thus, let us create the step by step procedure on how to import the data from MySQL to HDFS via SQOOP. MariaDB instead of MySQL which is sister branch of MySQL as an open source project. Thus, the commands in both are the same in MariaDB as well as in MySQL. Practical Aspects – Create Database – Use Database – Create table – Insert rows – View Table

Map Reduce:

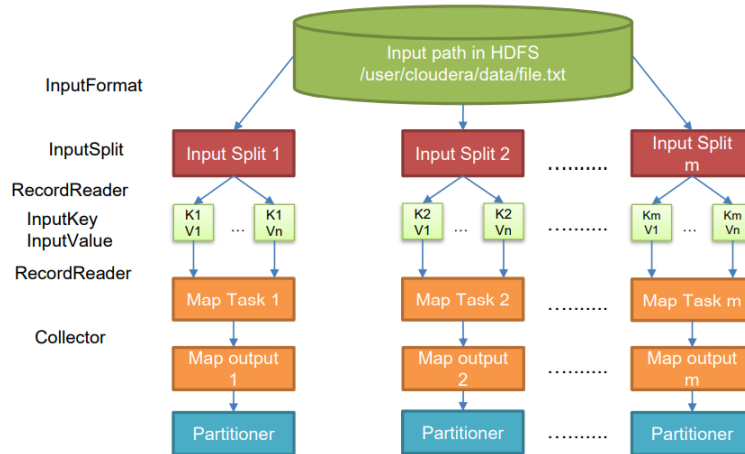


Fig 6: Concept of Mapper

The number of maps is usually driven by the total size of the inputs, that is, the total number of blocks of the input files. The right level of parallelism for maps seems to be around 10-100 maps per-node, although it has been set up to 300 maps for very CPU-light map tasks. Task setup takes a while, so it is best if the maps take at least a minute to execute. Thus, if you expect 10TB of input data and have a block size of 128MB, you'll end up with 82,000 maps.

Mapper= {(total data size)/ (input split size)}

Data size is 1 TB and input split size is 100 MB.

Mapper= (1000*1000)/100= 10,000

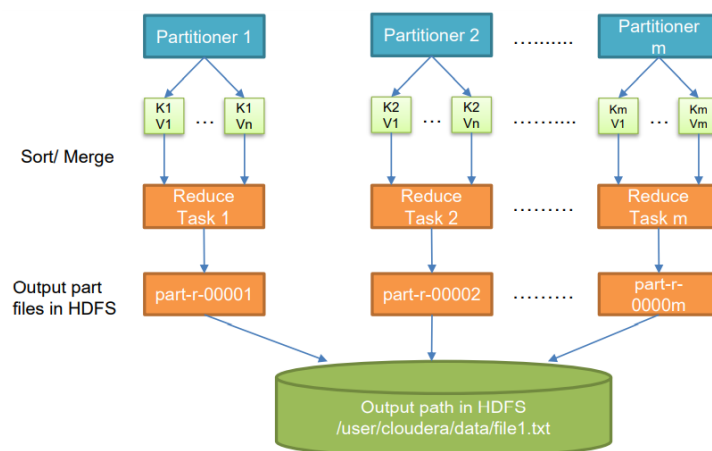


Fig 7: Concept of reducer

Big Data Processing, Management and Analytics

Data Ingestion:

The process of data ingestion — preparing data for analysis — usually includes steps called ☐ extract (taking the data from its current location), ☐ transform (cleansing and normalizing the data), ☐ load (placing the data in a database where it can be analyzed).

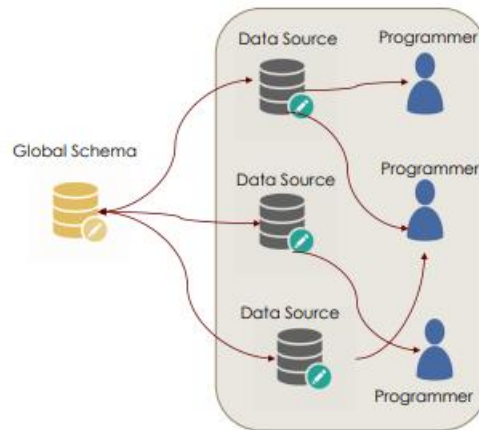


Fig 8: Automate Data Ingestion

Automate Data Ingestion:

Data has gotten too large, both in size and variety, to be curated manually. Thus, rather than manually defining a table's metadata, e.g., its schema or rules about minimum and maximum valid values, a user should be able to define this information in a spreadsheet, which is then read by a tool that enforces the specified metadata. This type of automation, by itself, can reduce the burden of data ingestion. But, in many cases, it does not eliminate the ingestion bottleneck, given the sheer number of tables involved. When thousands of tables must be ingested, filling out thousands of spreadsheets is better than writing thousands of ingestion scripts. However, it is still not a scalable or manageable task.

Govern the data to keep it clean:

Introducing data governance with a data steward responsible for the quality of each data source. Responsibility Includes: Defining Schema, cleansing rules, decision to ingest particular data into data source, treatment of dirty data. Besides data quality we also have data security and compliance with regulatory standards such as GDPR and master data management.

Advertise your cleansed data:

Once data source is cleansed, will other users be able to find it easily? If the data is point-to-point as requested, that data might not be useful to other customers and different people in the pool. Organization should implement a pub-sub (publish-subscribe) model with a registry of previously cleansed data available for lookup by all other users.

Why Mongo Database:

Reason 1: Aggregation framework:

It consists of 2 different varieties of Sets and they are

Map(): It performs operations like filtering the data and then performing sorting on that dataset.

Reduce(): It performs the operation of summarizing all the data after the map() operation.

Reason 2: BSON Format

It is JSON-like storage a format. BSON stands for Binary JSON. We can add data types like date and binary (JSON doesn't support). BSON format makes use of `_id` as a primary key over here. As stated that `_id` is being used as a primary key so it is having a unique value associated with itself called as Object Id, which is either generated by application driver or MongoDB service.

Reason 3: Sharding

The major problem with any web/mobile application is scaling. To overcome this MongoDB has added sharding feature. It is a method in which, data is being distributed across multiple machines. Horizontal scalability is being provided with the shard.

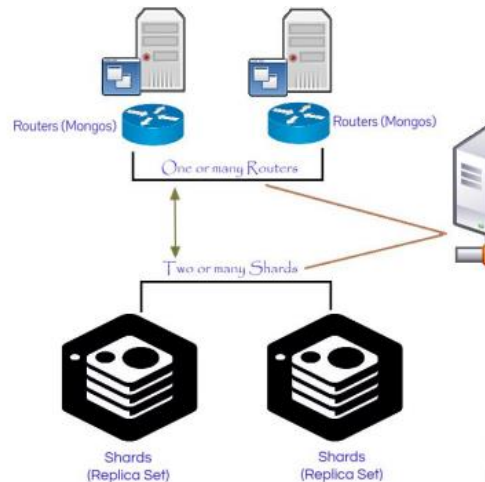


Fig 9: Sharding

Reason 4: Ad hoc queries

MongoDB can support ad hoc queries by using a unique query language or by indexing BSON documents. As it is a schema-less database(written in C++), it is much more flexible than the traditional database. Due to this, the data does not require much to set up for itself and reduced friction with OOP.

Reason 5: Schema-Less

The schema of a database describes the structure of the data to be stored. In a relational database, the schema defines its tables, its fields in each table and the relationships between each field and each table. The data stored needs to comply with the structure defined (tables, columns, data types and relations). So every register in a table have the same number of columns and format.

Reason 6: Capped Collections

MongoDB supports capped collection, as it is having fixed size of collections in it. It maintains the insertion order. Once the limit is reached it starts behaving like a circular queue. Example – Limiting our capped collection to 2MB `db.createCollection('logs', {capped: true, size: 2097152})`.

Reason 7: Indexing

To improve the performance of searches indexes are being created. We can index any field in MongoDB document either primary or secondary. Due to this reason, the database engine can efficiently resolve queries.

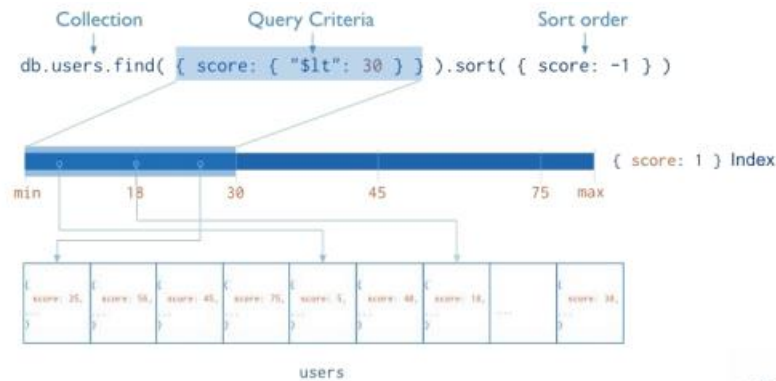


Fig 9: Indexing

Reason 8: Replication

Replication is being provided by distributing data across different machines. It can have one primary node and more than one secondary nodes in it (replica set). This set acts like a master-slave. Here, a master can perform read and write and a slave copies data from a master as a backup only for a read operation.²⁶

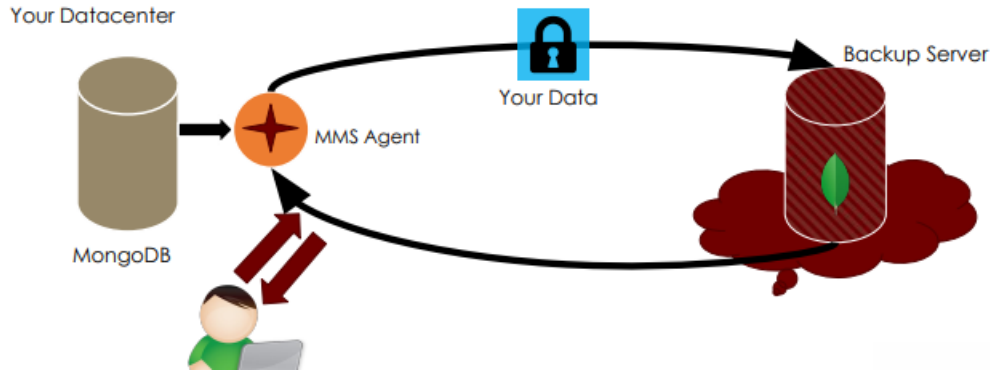


Fig 10: Mongo DB Management Service

5 CONCLUSION

To summarise the above presentation report, as a Vellore Institute of Technology student, the course's outcome has provided me with more content and knowledge of Bigdata. The goal of this course is to finish it and take a certificated exam on the 'FutureSkillsprime' portal. In this new field of computer science, I gained a vast amount of knowledge. Big data is a sea of information, and I can see millions of hours of work ahead of me in the future. I learned the basic fundamentals of bigdata in this course, which include the uses, causes, and benefits of using it. The Hadoop concept allowed for a greater understanding of big data. The vast concept in this course. Which took me more duration to complete certain amount this Hadoop concept. The installation and concept of using became very tedious job to learn it. Later after the completion of Hadoop moved on the topics not in detail but base idea was learnt from data ingestion and data pipelines.

REFERENCES

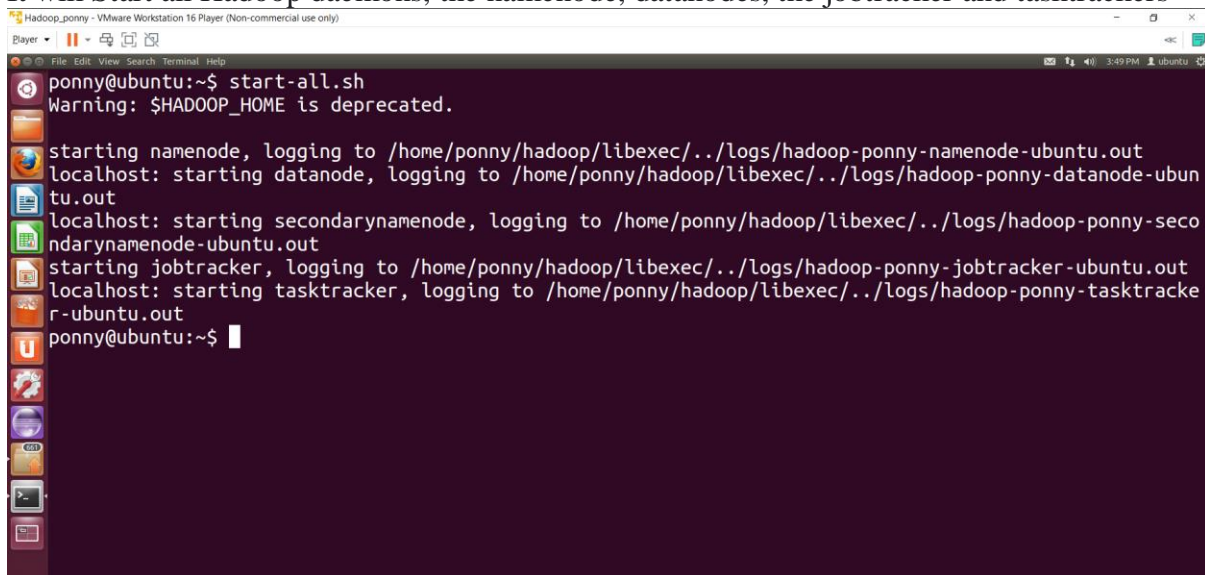
- Listed by Alphabetical order.
- All references should be cited in the document without miss
- Format

[1] L. Xiao, C. Xie, M. Min and W. Zhuang, "User-Centric View of Unmanned Aerial Vehicle Transmission Against Smart Attacks", *IEEE Transactions on Vehicular Technology*, vol. 67, no. 4, pp. 3420-3430, April 2018.

APPENDIX

>> start-all.sh

It will Start all Hadoop daemons, the namenode, datanodes, the jobtracker and tasktrackers



The screenshot shows a terminal window titled "Hadoop_ponny - VMware Workstation 16 Player (Non-commercial use only)". The terminal output is as follows:

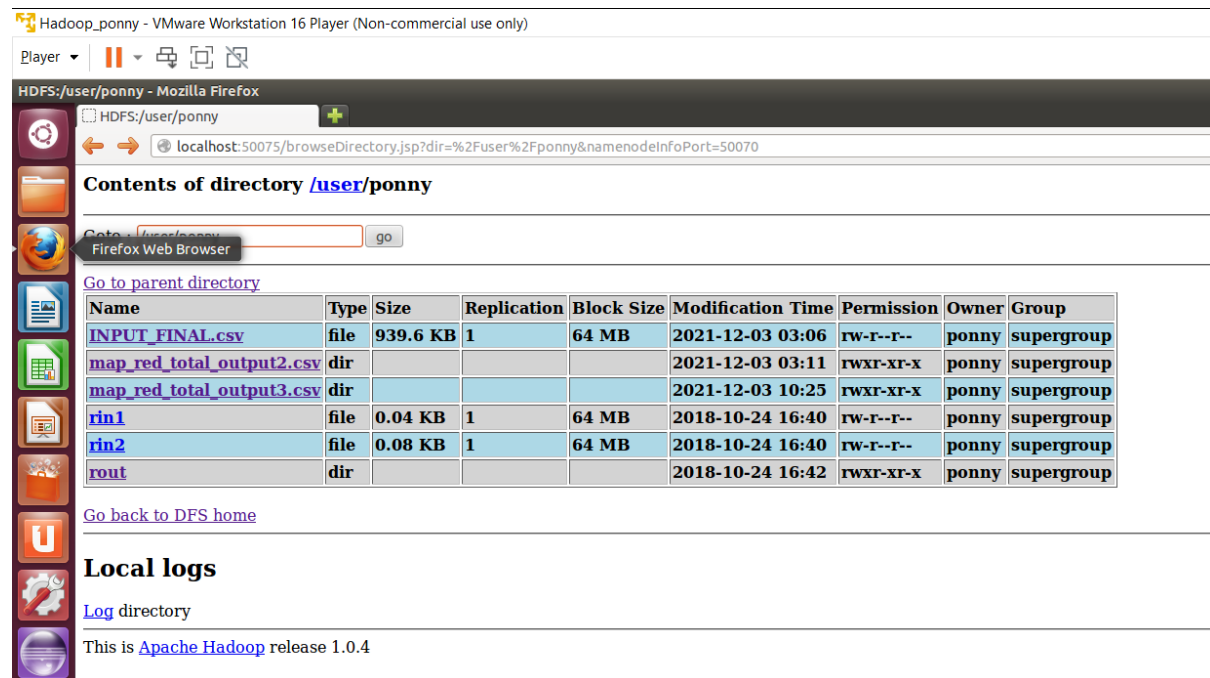
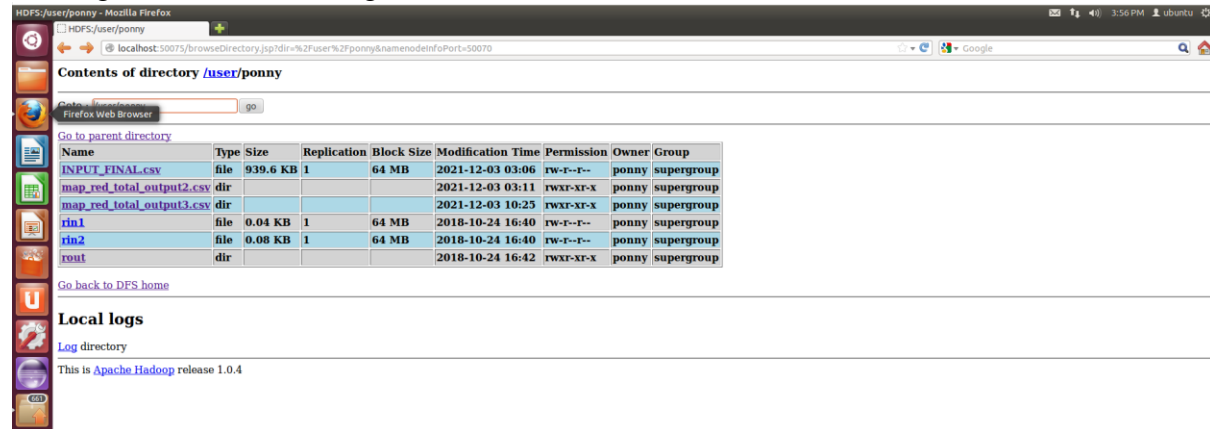
```
ponny@ubuntu:~$ start-all.sh
Warning: $HADOOP_HOME is deprecated.
starting namenode, logging to /home/ponny/hadoop/libexec/../logs/hadoop-ponny-namenode-ubuntu.out
localhost: starting datanode, logging to /home/ponny/hadoop/libexec/../logs/hadoop-ponny-datanode-ubuntu.out
localhost: starting secondarynamenode, logging to /home/ponny/hadoop/libexec/../logs/hadoop-ponny-secondarynamenode-ubuntu.out
starting jobtracker, logging to /home/ponny/hadoop/libexec/../logs/hadoop-ponny-jobtracker-ubuntu.out
localhost: starting tasktracker, logging to /home/ponny/hadoop/libexec/../logs/hadoop-ponny-tasktracker-ubuntu.out
ponny@ubuntu:~$
```

```
>> jps
```

It will check all the Hadoop daemons like DataNode, NodeManager, NameNode and ResourceManager

```
ponny@ubuntu:~$ jps
2675 NameNode
    System Settings    _ondaryNameNode
3511 TaskTracker
3263 JobTracker
2926 DataNode
3722 Jps
ponny@ubuntu:~$
```

Starting HDFS and checking the files:



Inserting the data in the hdfs

```
>>Hadoop fs -copyFromLocal covid_data.csv covid_data.csv
```

```
ponny@ubuntu: ~/Desktop$ hadoop fs -copyFromLocal covid_data.csv covid_data.csv
Warning: $HADOOP_HOME is deprecated.

ponny@ubuntu:~/Desktop$
```

The covid_data.csv file uploaded

Hadoop_ponny - VMware Workstation 16 Player (Non-commercial use only)

Player

HDFS:/user/ponny - Mozilla Firefox

localhost:50075/browseDirectory.jsp?dir=%2Fuser%2Fponny&namenodeinfoPort=50070

Contents of directory /user/ponny

Goto : /user/ponny

Go to parent directory

Name	Type	Size	Replication	Block Size	Modification Time	Permission	Owner	Group
INPUT_FINAL.csv	file	939.6 KB	1	64 MB	2021-12-03 03:06	rw-r--r--	ponny	supergroup
covid_data.csv	file	241.59 KB	1	64 MB	2021-07-29 16:25	rw-r--r--	ponny	supergroup
map_red_total_output2.csv	dir				2021-12-03 03:11	rw-r--r--	ponny	supergroup
map_red_total_output3.csv	dir				2021-12-03 10:25	rw-r--r--	ponny	supergroup
rin1	file	0.04 KB	1	64 MB	2018-10-24 16:40	rw-r--r--	ponny	supergroup
rin2	file	0.08 KB	1	64 MB	2018-10-24 16:40	rw-r--r--	ponny	supergroup
root	dir				2018-10-24 16:42	rw-r--r--	ponny	supergroup

Go back to DFS home

Local logs

Log directory

This is Apache Hadoop release 1.0.4

Hadoop_ponny - VMware Workstation 16 Player (Non-commercial use only)

Player ▾ | [Icons]

HDFS:/user/ponny - Mozilla Firefox

localhost:50075/browseDirectory.jsp?dir=%2Fuser%2Fponny&namenodeInfoPort=50070

Contents of directory /user/ponny

Goto : /user/ponny go

[Go to parent directory](#)

Name	Type	Size	Replication	Block Size	Modification Time	Permission	Owner	Group
INPUT_FINAL.csv	file	939.6 KB	1	64 MB	2021-12-03 03:06	rw-r--r--	ponny	supergroup
covid_data.csv	file	241.59 KB	1	64 MB	2021-07-29 16:25	rw-r--r--	ponny	supergroup
map_red_total_output2.csv	dir				2021-12-03 03:11	rw-r--r--	ponny	supergroup
map_red_total_output3.csv	dir				2021-12-03 10:25	rw-r--r--	ponny	supergroup
rin1	file	0.04 KB	1	64 MB	2018-10-24 16:40	rw-r--r--	ponny	supergroup
rin2	file	0.08 KB	1	64 MB	2018-10-24 16:40	rw-r--r--	ponny	supergroup
rout	dir				2018-10-24 16:42	rw-r--r--	ponny	supergroup

[Go back to DFS home](#)

Local logs

[Log directory](#)

This is [Apache Hadoop](#) release 1.0.4

Hadoop map reduce code to calculate the number of covid cases at each country

Hadoop_ponny - VMware Workstation 16 Player (Non-commercial use only)

Player ▾ | [Icons]

File Edit Source Refactor Navigate Search Project Run Window Help

Project Explorer

- bda_crime
 - src
 - map_reduce_codes
 - map_red_total.java
 - noofcases.java**

JRE System Library [JavaSE-1.7]

- resources.jar - /usr/lib/jvm/java-7-openjdk-i386/jre/lib
- rt.jar - /usr/lib/jvm/java-7-openjdk-i386/jre/lib
- jsse.jar - /usr/lib/jvm/java-7-openjdk-i386/jre/lib
- jce.jar - /usr/lib/jvm/java-7-openjdk-i386/jre/lib
- charsets.jar - /usr/lib/jvm/java-7-openjdk-i386/jre/lib
- rhino.jar - /usr/lib/jvm/java-7-openjdk-i386/jre/lib
- pulse-java.jar - /usr/lib/jvm/java-7-openjdk-i386/jre/lib
- java-atk-wrapper.jar - /usr/lib/jvm/java-7-openjdk-i386/jre/lib
- org.GNOME.Accessibility
- META-INF
- zipfs.jar - /usr/lib/jvm/java-7-openjdk-i386/jre/lib/ext
- sunjce_provider.jar - /usr/lib/jvm/java-7-openjdk-i386/jre/lib/ext
- dnsns.jar - /usr/lib/jvm/java-7-openjdk-i386/jre/lib/ext
- sunpkcs11.jar - /usr/lib/jvm/java-7-openjdk-i386/jre/lib/ext
- localedata.jar - /usr/lib/jvm/java-7-openjdk-i386/jre/lib/ext
- hadoop-ant-1.0.4.jar - /home/ponny/hadoop
- hadoop-client-1.0.4.jar - /home/ponny/hadoop
- hadoop-cookbook-chapter8.jar - /home/ponny/hadoop
- hadoop-core-1.0.4.jar - /home/ponny/hadoop
- hadoop-examples-1.0.4.jar - /home/ponny/hadoop
- hadoop-minicuster-1.0.4.jar - /home/ponny/hadoop
- hadoop-test-1.0.4.jar - /home/ponny/hadoop

noofcases.java

```
package map_reduce_codes;

import java.io.IOException;
import java.util.*;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.conf.*;
import org.apache.hadoop.io.*;
import org.apache.hadoop.mapreduce.*;
import org.apache.hadoop.mapreduce.lib.input.*;
import org.apache.hadoop.mapreduce.lib.output.*;

public class noofcases {

    public static class Map extends Mapper < LongWritable, Text, Text, IntWritable > {
        IntWritable one = new IntWritable(1);
        Text new key = new Text();

        public void map(LongWritable key, Text value, Context context) throws IOException,
            InterruptedException {
            String[] line = value.toString().split(",");
            context.write(new Text(line[1]), new IntWritable(Integer.parseInt(line[2])));
        }
    }

    public static class Reduce extends Reducer < Text, IntWritable, Text, IntWritable > {

        public void reduce(Text key, Iterable < IntWritable > values, Context context)
            throws IOException,
            InterruptedException {
            int total = 0;
            for (IntWritable val: values) {
                total+=val.get();
            }
            context.write(key,new IntWritable(total));
        }
    }
}
```

```

map_red_total.java  noofcases.java
package map_reduce_codes;

import java.io.IOException;
import java.util.*;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.conf.*;
import org.apache.hadoop.io.*;
import org.apache.hadoop.mapreduce.*;
import org.apache.hadoop.mapreduce.lib.input.*;
import org.apache.hadoop.mapreduce.lib.output.*;

public class noofcases {
    public static class Map extends Mapper < LongWritable, Text, Text, IntWritable > {
        IntWritable one = new IntWritable(1);
        Text new_key = new Text();
        public void map(LongWritable key, Text value, Context context) throws IOException,
        InterruptedException {
            String[] line = value.toString().split(",");
            context.write(new Text(line[1]), new IntWritable(Integer.parseInt(line[2])));
        }
    }

    public static class Reduce extends Reducer < Text, IntWritable, Text, IntWritable > {

        public void reduce(Text key, Iterable < IntWritable> values, Context context)
        throws IOException,
        InterruptedException {
            int total = 0;
            for (IntWritable val: values) {
                total+=val.get();
            }
            context.write(key,new IntWritable(total));
        }
    }
}

```

>> start-all.sh

It will start all the Hadoop daemons,the namenode ,datanode,the job tracker and the task tracker

```

Hadoop_ponny - VMware Workstation 16 Player (Non-commercial use only)
Player
ponny@ubuntu:~/Desktop$
ponny@ubuntu:~/Desktop$ stop-all.sh
Warning: $HADOOP_HOME is deprecated.

stopping jobtracker
localhost: stopping tasktracker
stopping namenode
localhost: stopping datanode
localhost: stopping secondarynamenode
ponny@ubuntu:~/Desktop$ start-all.sh
Warning: $HADOOP_HOME is deprecated.

starting namenode, logging to /home/ponny/hadoop/libexec/../logs/hadoop-ponny-namenode-ubuntu.out
localhost: starting datanode, logging to /home/ponny/hadoop/libexec/../logs/hadoop-ponny-datanode-ubuntu.out
localhost: starting secondarynamenode, logging to /home/ponny/hadoop/libexec/../logs/hadoop-ponny-secondarynamenode-ubuntu.out
starting jobtracker, logging to /home/ponny/hadoop/libexec/../logs/hadoop-ponny-jobtracker-ubuntu.out
localhost: starting tasktracker, logging to /home/ponny/hadoop/libexec/../logs/hadoop-ponny-tasktracker-ubuntu.out
ponny@ubuntu:~/Desktop$

```

>>jps

It will check wheather all the Hadoop deamons mentioned above are running


```
ponny@ubuntu:~/Desktop$ jps
6485 TaskTracker
6150 SecondaryNameNode
5896 DataNode
6232 JobTracker
6594 Jps
5650 NameNode
4519
ponny@ubuntu:~/Desktop$ hadoop jar /home/ponny/cs.jar map_reduce_codes.noofcases covid_data.csv ANS_YEAR_WISE_COVID_DATA.csv
Warning: $HADOOP_HOME is deprecated.

21/07/29 18:39:19 WARN mapred.JobClient: Use GenericOptionsParser for parsing the arguments. Applications should implement Tool for the same.
21/07/29 18:39:19 INFO input.FileInputFormat: Total input paths to process : 1
21/07/29 18:39:19 INFO util.NativeCodeLoader: Loaded the native-hadoop library
21/07/29 18:39:19 WARN snappy.LoadSnappy: Snappy native library not loaded
21/07/29 18:39:19 INFO mapred.JobClient: Running job: job_202107291834_0001
21/07/29 18:39:20 INFO mapred.JobClient: map 0% reduce 0%
21/07/29 18:39:35 INFO mapred.JobClient: map 100% reduce 0%
21/07/29 18:39:50 INFO mapred.JobClient: map 100% reduce 100%
21/07/29 18:39:55 INFO mapred.JobClient: Job complete: job_202107291834_0001
21/07/29 18:39:55 INFO mapred.JobClient: Counters: 29
21/07/29 18:39:55 INFO mapred.JobClient: Job Counters
21/07/29 18:39:55 INFO mapred.JobClient: Launched reduce tasks=1
```

>>Hadoop jar /home/ponny/cs.jar map_reduce_codes.noofcases covid_data.csv
ANS_YEAR_WISE_COVID_DATA.csv

```
Hadoop_ponny - VMware Workstation 16 Player (Non-commercial use only)
Player
ponny@ubuntu:~/Desktop$ jps
6485 TaskTracker
6150 SecondaryNameNode
5896 DataNode
6232 JobTracker
6594 Jps
5650 NameNode
4519
ponny@ubuntu:~/Desktop$ hadoop jar /home/ponny/cs.jar map_reduce_codes.noofcases covid_data.csv ANS_YEAR_WISE_COVID_DATA.csv
Warning: $HADOOP_HOME is deprecated.

21/07/29 18:39:19 WARN mapred.JobClient: Use GenericOptionsParser for parsing the arguments. Applications should implement Tool for the same.
21/07/29 18:39:19 INFO input.FileInputFormat: Total input paths to process : 1
21/07/29 18:39:19 INFO util.NativeCodeLoader: Loaded the native-hadoop library
21/07/29 18:39:19 WARN snappy.LoadSnappy: Snappy native library not loaded
21/07/29 18:39:19 INFO mapred.JobClient: Running job: job_202107291834_0001
21/07/29 18:39:20 INFO mapred.JobClient: map 0% reduce 0%
```

```
Hadoop_ponny - VMware Workstation 16 Player (Non-commercial use only)
Player
File Edit View Search Terminal Help
21/07/29 18:39:55 INFO mapred.JobClient: SLOTS_MILLIS_MAPS=10936
21/07/29 18:39:55 INFO mapred.JobClient: Total time spent by all reduces waiting after reserving s
lots (ms)=0
21/07/29 18:39:55 INFO mapred.JobClient: Total time spent by all maps waiting after reserving slot
s (ms)=0
21/07/29 18:39:55 INFO mapred.JobClient: Launched map tasks=1
21/07/29 18:39:55 INFO mapred.JobClient: Data-local map tasks=1
21/07/29 18:39:55 INFO mapred.JobClient: SLOTS_MILLIS_REDUCE=12499
21/07/29 18:39:55 INFO mapred.JobClient: File Output Format Counters
21/07/29 18:39:55 INFO mapred.JobClient: Bytes Written=2847
21/07/29 18:39:55 INFO mapred.JobClient: FileSystemCounters
21/07/29 18:39:55 INFO mapred.JobClient: FILE_BYTES_READ=146330
21/07/29 18:39:55 INFO mapred.JobClient: HDFS_BYTES_READ=247503
21/07/29 18:39:55 INFO mapred.JobClient: FILE_BYTES_WRITTEN=335837
21/07/29 18:39:55 INFO mapred.JobClient: HDFS_BYTES_WRITTEN=2847
21/07/29 18:39:55 INFO mapred.JobClient: File Input Format Counters
21/07/29 18:39:55 INFO mapred.JobClient: Bytes Read=247390
21/07/29 18:39:55 INFO mapred.JobClient: Map-Reduce Framework
21/07/29 18:39:55 INFO mapred.JobClient: Map output materialized bytes=146330
21/07/29 18:39:55 INFO mapred.JobClient: Map input records=9613
21/07/29 18:39:55 INFO mapred.JobClient: Reduce shuffle bytes=146330
21/07/29 18:39:55 INFO mapred.JobClient: Spilled Records=19226
21/07/29 18:39:55 INFO mapred.JobClient: Map output bytes=127098
21/07/29 18:39:55 INFO mapred.JobClient: Total committed heap usage (bytes)=177016832
21/07/29 18:39:55 INFO mapred.JobClient: CPU time spent (ms)=780
21/07/29 18:39:55 INFO mapred.JobClient: Combine input records=0
21/07/29 18:39:55 INFO mapred.JobClient: SPLIT_RAW_BYTES=113
21/07/29 18:39:55 INFO mapred.JobClient: Reduce input records=9613
21/07/29 18:39:55 INFO mapred.JobClient: Reduce input groups=205
21/07/29 18:39:55 INFO mapred.JobClient: Combine output records=0
21/07/29 18:39:55 INFO mapred.JobClient: Physical memory (bytes) snapshot=196513792
21/07/29 18:39:55 INFO mapred.JobClient: Reduce output records=205
21/07/29 18:39:55 INFO mapred.JobClient: Virtual memory (bytes) snapshot=767225856
21/07/29 18:39:55 INFO mapred.JobClient: Map output records=9613
ponny@ubuntu:~/Desktop$
```

Check file ANS_YEAR_WISE_COVID_DATA.csv

Hadoop_ponny - VMware Workstation 16 Player (Non-commercial use only)

Player

File Edit View History Bookmarks Tools Help

HDFS:/user/ponny

localhost:50075/browseDirectory.jsp?dir=/user/ponny&namenodeInfoPort=50070

Contents of directory /user/ponny

Goto : /user/ponny go

[Go to parent directory](#)

Name	Type	Size	Replication	Block Size	Modification Time	Permission	Owner	Group
ANS_YEAR_WISE_COVID_DATA.csv	dir				2021-07-29 18:39	rw-r--r--	ponny	supergroup
INPUT_FINAL.csv	file	939.6 KB	1	64 MB	2021-12-03 03:06	rw-r--r--	ponny	supergroup
covid_data.csv	file	241.59 KB	1	64 MB	2021-07-29 16:25	rw-r--r--	ponny	supergroup
map_red_total_output2.csv	dir				2021-12-03 03:11	rw-r--r--	ponny	supergroup
map_red_total_output3.csv	dir				2021-12-03 10:25	rw-r--r--	ponny	supergroup
rin1	file	0.04 KB	1	64 MB	2018-10-24 16:40	rw-r--r--	ponny	supergroup
rin2	file	0.08 KB	1	64 MB	2018-10-24 16:40	rw-r--r--	ponny	supergroup
rout	dir				2018-10-24 16:42	rw-r--r--	ponny	supergroup

[Go back to DFS home](#)

Local logs

[Log](#) directory

This is [Apache Hadoop](#) release 1.0.4

Hadoop_ponny - VMware Workstation 16 Player (Non-commercial use only)

Player

HDFS:/user/ponny/ANS_YEAR_WISE_COVID_DATA.csv/part-r-00000 - Mozilla Firefox

HDFS:/user/ponny/ANS_YEAR...

localhost:50075/browseBlock.jsp?blockId=1604199890738157153&blockSize=2847&genstamp=1503&filename=%2Fuser%2Fponny%2FANS_YEAR_WISE_COVID_DATA.csv%2Fpart-r-00000

File: /user/ponny/ANS_YEAR_WISE_COVID_DATA.csv/part-r-00000

Goto : /user/ponny/ANS_YEAR_WISE_CO go

[Go back to dir listing](#)
[Advanced view/download options](#)

Afghanistan	367
Albania	383
Algeria	1468
Andorra	545
Angola	17
Anguilla	3
Antigua and Barbuda	15
Argentina	1715
Armenia	853
Aruba	74
Australia	5956
Austria	12640
Azerbaijan	717
Bahamas	36
Bahrain	811
Bangladesh	164
Barbados	63
Belarus	861
Belgium	22194
Belize	7
Benin	26
Bermuda	39
Bhutan	5
Bolivia	210
Bonaire Sint Eustatius and Saba	2
Bosnia and Herzegovina	781

[Download this file](#)
[Tail this file](#)

Chunk size to view (in bytes, up to file's DFS block size):

Total number of blocks: 1
1604199890738157153: [127.0.0.1:50010](#)

Output of the csv File

Afghanistan	367	
Albania	383	
Algeria	1468	
Andorra	545	
Angola	17	
Anguilla	3	
Antigua and Barbuda	15	
Argentina	1715	
Armenia	853	
Aruba	74	
Australia	5956	
Austria	12640	
Azerbaijan	717	
Bahamas	36	
Bahrain	811	
Bangladesh	164	
Barbados	63	
Belarus	861	
Belgium	22194	
Belize	7	
Benin	26	
Bermuda	39	
Bhutan	5	
Bolivia	210	
Bonaire Sint Eustatius and Saba	2	
Bosnia and Herzegovina	781	
