UNIT VI : **Data Visualization and Hadoop**

- Introduction to Data Visualization, Challenges to Big data visualization, Types of data visualization, Data Visualization Techniques, Visualizing Big Data, Tools used in Data Visualization, Hadoop ecosystem, Map Reduce, Pig, Hive, Analytical techniques used in Big data visualization, Data visualization with Tableau. Data Visualization using Python : Line plot, scatter plot, Histogram, density plot, Box- plot

Introduction to Data Visualization

- Data visualization is the study of representing data or information in a visual form. Visuals come in the different forms like graphs, images, diagrams, or animations etc.
- Visualization is an excellent medium to analyze, comprehend, and share the information.

Big Data Visualization

- Data visualization is representing data in some systematic form including attributes and variables for the unit of information.
- Visualization is an important approach to helping Big Data get a complete view of data and discover data values.
- Big data are high volume, high velocity, and/or high variety datasets that require new forms of processing to enable enhanced process optimization, insight discovery and decision making. Challenges of Big Data lie in data capture, storage, analysis, sharing, searching, and visualization

- Advantages of visualization:
- Visual images help to transmit a huge amount of information to the human brain at a glance.
- Visual images help in establishing relationships and distinction between different patterns, or processes easily.
- Visual interpretation help in exploring data from different angles, which help gain insights.
- Visualization helps in identifying problems and understanding trends and outliers.
- Visualization point out key or interesting breakthroughs in a large dataset.

- Benefits of visualization :

- Improved decision making
- Better adhoc data analysis
- Improved collaboration/information sharing
- Provide self-service capabilities to end users
- Increased return on investment (ROI)
- Time saving
- Reduced burden on IT

Types of Data Visualisation

Name	Description	Tool
1D/Linear	List of Items	No tool is used
2D/Planar	cartogram, dot distribution map	Google maps API
3D/Volumetric	3D computer Models	AutoQ3D, TrueSpace
Temporal	Timeline , Time series Time Flow, Gantt chart	Excel Timeplot, Google Charts
Mutidimentional	pie chart, Histogram, bar chart,	Google Charts, Tableau Public
Tree/Hierachical	Dendogram, Wedge Stack graph	Sci2, Google Charts. Network workbench
Network	Matrix, Node LInk diagram	Google Fusion Tables

- Techniques used for visual data Representation :

- ISOLINE,
- Isosurface,
- Direct volume rendering,
- Streamline, Map,
- Parallel coordinate plot,
- Venn Diagram,
- Time line,
- Euler Diagram,
- Hyperbolic trees,
- Cluster Diagram,

- Applications of Data Visualisation :

- Education
- Information
- Production
- Science
- Systems Visualisation
- Visual Communication
- Visual analytics.

- Tools Used in Data Visualisation :
- Excel
- Last.Forward,
- Digg.Com
- Pics,
- Arc
- Google Chrats API
- Twittearth
- Tag Galaxy
- Tableau

- **Challenges of Big Data Visualization**
- Scalability and dynamics are two major challenges in visual analytics.
- The visualization-based methods take the challenges presented by the "four Vs" of big data and turn them into following opportunities.
- *Volume*: The methods are developed to work with an immense number of datasets and enable to derive meaning from large volumes of data.
- *Variety*: The methods are developed to combine as many data sources as needed.
- *Velocity*: With the methods, businesses can replace batch processing with real-time stream processing.
- *Value*: The methods not only enable users to create attractive infographics and heatmaps, but also create business value by gaining insights from big data.

- **Visualization of big data with diversity and heterogeneity (structured, semi-structured, and unstructured) is a big problem.**
- **Speed is the desired factor for the big data analysis. Designing a new visualization tool with efficient indexing is not easy in big data.**
- **Cloud computing and advanced graphical user interface can be merged with the big data for the better management of big data scalability.**
- **Visualization systems must contend with unstructured data forms such as graphs, tables, text, trees, and other metadata.**
- **Big data often has unstructured formats. Due to bandwidth limitations and power requirements, visualization should move closer to the data to extract meaningful information efficiently.**
- **Because of the big data size, the need for massive parallelization is a challenge in visualization.**
- **The challenge in parallel visualization algorithms is decomposing a problem into independent tasks that can be run concurrently**

- **Conventional Data Visualization Methods**
- Many conventional data visualization methods are often used. They are: table, histogram, scatter plot, line chart, bar chart, pie chart, area chart, flow chart, bubble chart, multiple data series or combination of charts, time line, Venn diagram, data flow diagram, and entity relationship diagram, etc. In addition, some data visualization methods have been used although they are less known compared the above methods. The additional methods are: parallel coordinates, treemap, cone tree, and semantic network, etc.

- Hadoop MapReduce- Java-based Processing Framework for Big Data
- MapReduce programming paradigm uses a two-step data analysis process- Map Stage and Reduce Stage (reduce phase is optional).
- The map stage takes a set of data and converts it into another set where data elements are broken down into key-value pairs or tuples. Reduce job takes the output of the map function and combines them into smaller set of tuples or key-value pairs.

- MapReduce Terminologies
- **Job** - It is the complete process to execute including the mappers, the input, the reducers and the output across a particular dataset.
- **Task** - Every job is divided into several mappers and reducers. A portion of the job executed on a slice of data can be referred to as a task.
- **JobTracker** - It is the master node for managing all the jobs and resources in a hadoop cluster.
- **TaskTracker** - These are the agents deployed to each machine in the hadoop cluster to run Map and Reduce tasks and then report the status to the JobTracker after execution.

- Hadoop WordCount operation occurs in 3 stages –
- Mapper Phase
- Shuffle Phase
- Reducer Phase

- **Mapper Phase**
- The text from the input text file is tokenized into words to form a key value pair with all the words present in the input text file. The key is the word from the input file and value is ‘1’.
- For Example the sentence “PICT is an engineering College”. The mapper phase in the WordCount example will split the string into individual tokens i.e. words. In this case, the entire sentence will be split into 5 tokens (one for each word) with a value 1.

- (PICT,1)
- (is,1)
- (an,1)
- (engineering,1)
- (College,1)

- Key-Value pairs from Hadoop Map Phase Execution-

- **Shuffle Phase Execution:**
- After the map phase execution is completed successfully, shuffle phase is executed automatically wherein the key-value pairs generated in the map phase are taken as input and then sorted in alphabetical order.
- (an,1)
- (College,1)
- (engineering,1)
- (is,1)
- (PICT,1)

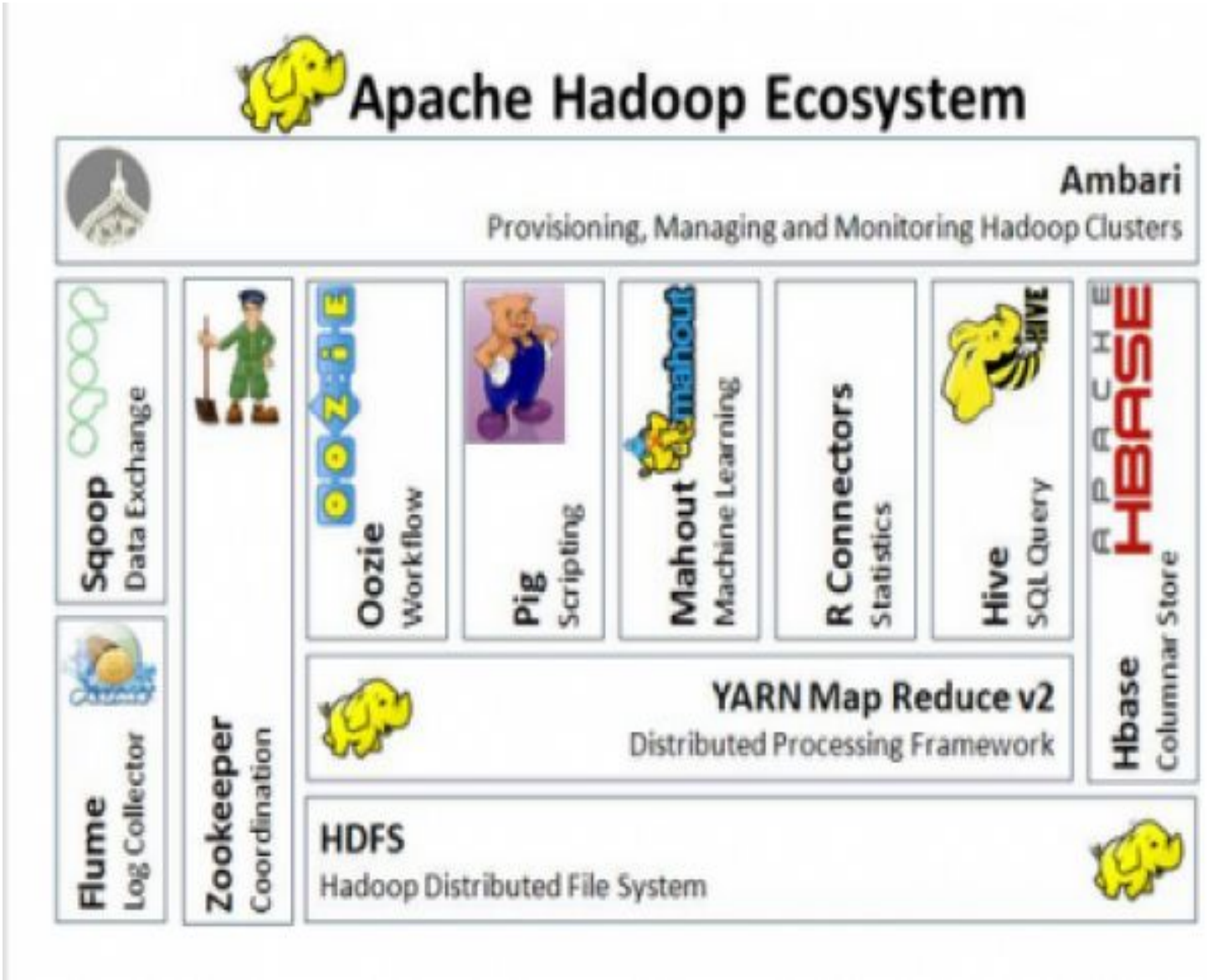
- **Reducer Phase Execution:**
- In the reduce phase, all the keys are grouped together and the values for similar keys are added up to find the occurrences for a particular word. It is like an aggregation phase for the keys generated by the map phase. The reducer phase takes the output of shuffle phase as input and then reduces the key-value pairs to unique keys with values added up.

- All Engineering colleges in Pune

- Output after reduce
- All 1
- Engineering 2
- Colleges 1
- college

- **Apache Hadoop framework is composed of the following modules**
- Hadoop Common: contains libraries and utilities needed by other Hadoop modules
- Hadoop Distributed File System (HDFS): a distributed file-system that stores data on the commodity machines, providing very high aggregate bandwidth across the cluster
- Hadoop YARN: a resource-management platform responsible for managing compute resources in clusters and using them for scheduling of users' applications.

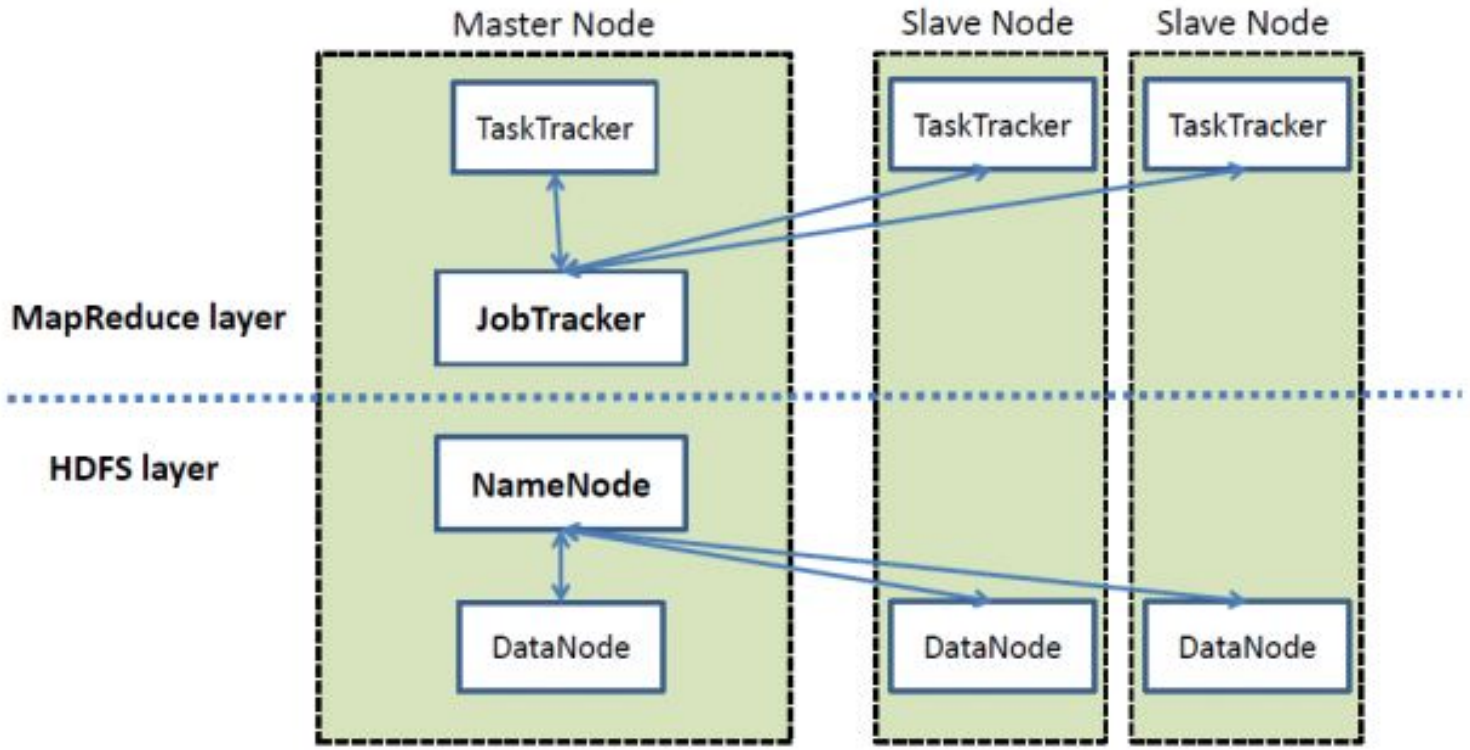
- Hadoop MapReduce: a programming model for large scale data processing.
- Apache Hadoop's MapReduce and HDFS components originally derived respectively from Google's MapReduce and Google File System (GFS) papers.
- Beyond HDFS, YARN and MapReduce, the entire Apache Hadoop "platform" is now commonly considered to consist of a number of related projects as well: Apache Pig, Apache Hive, Apache HBase, and others.


Apache Hadoop Ecosystem
Ambari
Provisioning, Managing and Monitoring Hadoop Clusters
Sqoop
Data Exchange
Flume
Log Collector
Zookeeper
Coordination
Oozie
Workflow
Pig
Scripting
Mahout
Machine Learning
R Connectors
Statistics
Hive
SQL Query
Hbase
Columnar Store
YARN Map Reduce v2
Distributed Processing Framework
HDFS
Hadoop Distributed File System

- Apache Pig and Apache Hive, among other related projects, expose higher level user interfaces like Pig latin and a SQL variant respectively. The Hadoop framework itself is mostly written in the Java programming language, with some native code in C and command line utilities written as shell-scripts.

- **HDFS and MapReduce**
- There are two primary components at the core of Apache Hadoop.
- the Hadoop Distributed File System (HDFS) and the MapReduce parallel processing framework. These are both open source projects, inspired by technologies created inside Google.

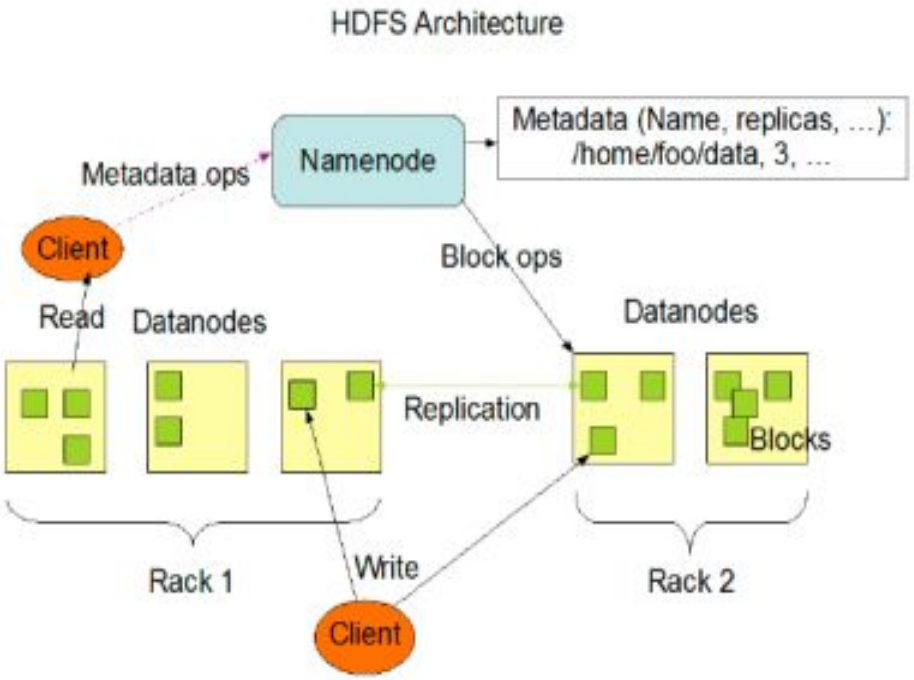
High Level Architecture of Hadoop



- **Hadoop distributed file system**
- The Hadoop distributed file system (HDFS) is a distributed, scalable, and portable file-system written in Java for the Hadoop framework. Each node in a Hadoop instance typically has a single namenode, and a cluster of datanodes form the HDFS cluster. The situation is typical because each node does not require a datanode to be present. Each datanode serves up blocks of data over the network using a block protocol specific to HDFS. The file system uses the TCP/IP layer for communication. Clients use Remote procedure call (RPC) to communicate between each other.

HDFS Terminology

- Namenode
- Datanode
- DFS Client
- Files/Directories
- Replication
- Blocks
- Rack-awareness



- https://opensource.com/life/14/8/intro-apache-hadoop-big-data

- hadoop jar
- *The* hadoop jar *command runs a program contained in a JAR file. Users can bundle their MapReduce code in a JAR file and execute it using this command.*
- **Syntax**
- hadoop jar <jar> [<arguments>]
- **Running from a JAR file:**
- The simple Word Count program is another example of a program that is run using the hadoop jar command. The wordcount functionality is built into the hadoop-*examples.jar program.
- We have to pass the file, along with the location, to Hadoop with the hadoop jar command and Hadoop reads the JAR file and executes the relevant instructions.
- The Word Count program reads files from an input directory, counts the words, and writes the results of the application to files in an output directory.

- https://hadoop.apache.org/docs/stable/hadoop-mapreduce-client/hadoop-mapreduce-client-core/MapReduceTutorial.html

- The WordCount application :
- The Mapper implementation, via the map method, processes one line at a time, as provided by the specified TextInputFormat. It then splits the line into tokens separated by whitespaces, via the StringTokenizer, and emits a key-value pair of < <word>, 1>
- < Hello, 1>
- < World, 1>
- < Bye, 1>
- < World, 1>

- public void map(Object key, Text value, Context context) throws IOException, InterruptedException {

  StringTokenizer itr = new StringTokenizer(value.toString());

  while (itr.hasMoreTokens())

  {

  word.set(itr.nextToken());

  context.write(word, one);

  } }

- WordCount also specifies a combiner. Hence, the output of each map is passed through the local combiner (which is same as the Reducer as per the job configuration) for local aggregation, after being sorted on the *key*s.
- job.setCombinerClass(IntSumReducer.class);

- The Reducer implementation, via the reduce method just sums up the values, which are the occurrence counts for each key (i.e. words in this example).
- public void reduce(Text key, Iterable<IntWritable> values, Context context) throws IOException, InterruptedException

  { int sum = 0; for (IntWritable val : values)

  { sum += val.get(); }

  result.set(sum);

  context.write(key, result);

  }