Big Data Analytics

Unit – 1 : Introduction to Big Data : Big data features and challenges, Problems with Traditional Large-Scale System , Sources of Big Data, 3 V's of Big Data, Types of Data.

Working with Big Data : Google File System. Hadoop Distributed.

File System (HDFS) : Building blocks of Hadoop (Namenode, Data node, Secondary Namenode, Job Tracker, Task Tracker), Introducing and Configuring Hadoop cluster (Local, Pseudodistributed mode, Fully Distributed mode), Configuring XML files.

Unit – 2 : Writing MapReduce Programs : A Weather Dataset, Understanding Hadoop API for MapReduce Framework (Old and New), Basic programs of Hadoop MapReduce: Driver code, Mapper code, Reducer code, Record Reader, Combiner, Partitioner.

Unit – 3 : Hadoop I/O : The Writable Interface, Writable Comparable and comparators.

Writable Classes : Writable wrappers for Java primitives. Text, Bytes Writable, Null Writable, Object Writable and Generic Writable. Writable collections, Implementing a Custom.

Writable : Implementing a Raw Comparator for speed, Custom comparators.

Unit – 4 : Pig : Hadoop Programming Made Easier Admiring the Pig Architecture, Going with the Pig Latin Application Flow, Working through the ABCs of Pig Latin, Evaluating Local and Distributed Modes of Running Pig Scripts, Checking out the Pig Script Interfaces, Scripting with Pig Latin.

Unit – 5 : Applying Structure to Hadoop Data with Hive : Saying Hello to Hive, Seeing How the Hive is Put Together, Getting Started with Apache Hive, Examining the Hive Clients, Working with Hive Data Types. Creating and Managing Databases and Tables, Seeing How the Hive Data Manipulation Language Works, Querying and Analyzing Data.

Disaster Management

Unit – 1 : Understanding Disasters and Hazards and related issues social and environmental. Risk and Vulnerability. Types of Disasters, their occurrence/ causes, impact and preventive measures.

Unit – 2 : Natural Disasters : Hydro-meteorological Based Disasters like Flood, Flash Flood, Cloud Burst, Drought, Cyclone, Forest Fires; Geological Based Disasters like Earthquake, Tsunami, Landslides, Volcanic Eruptions.

Unit – 3 : Man made Disasters : Textile Processing Industrial Hazards, Major Power Break Downs, Traffic Accidents, Fire Hazards.

Unit – 4 : Management roll in mitigating Disaster in Indian Textile Industries. Roll of production people in Disaster Management.

Energy Management

Unit – 1 : Energy Basics : Energy Demand Management Conservation & Resource Development, Energy for Sustainabl Development.

Unit – 2 : Need for Energy Management by Sector : Indust Buildings & Houses, Transport, Electric Power.

Unit – 3 : Need for Energy Management by Secto Agriculture, Domestic; Energy forecasting techniques; Ener Integration, Energy Matrix.

Unit – 4 : Energy Auditing : Energy management for clea production, application of renewable energy, appropr technologies.

Finite Element Methods

Unit – 1 : Review of Mathematics : Introduction to FEM its applications. Advantages of FEM, comparison with methods such as FDM and FVM. Review of matrix algebra, elimination method, banded symmetric matrix and bandwidt

Unit – 2 : Discretization & Finite Element Formulation Governing Differential Equations : Geometrical approxim Element shapes and behaviour, Choice of element types, si number of elements, Location of nodes; p and h method o refinement; Shape functions and their properties; Assemb boundary conditions. General field problems, discre continuous models; Method of weighted residuals. Ga method and other methods; Introduction to variational form (Ritz technique); Convergence of solution, compatibility.

Unit – 3 : One-dimensional Finite Element Analys dimensional second order equation, derivation of shape fu Stiffness matrix and force vectors, assembly of elemental Derivation of finite elements equations using potentia approach, 1-D bar element. Longitudinal vibration and mod fourth order beam equation, transverse deflections an frequencies, solution of problems from fluid mechanics transfer.

Unit – 4 : Two-dimensional Finite Element Analy element formulation using three node triangular (CST) el four node rectangular element, Plane stress and Pl

BIG DATA ANALYTICS

VIII C.S.

1 INTRODUCTION TO BIG DATA

IMPORTANT QUESTIONS

PART-A

Q.1 ***What do you mean by web data?***

Ans. Web Data : Web data is the data present on web servers (or enterprise servers) in the form of text, images, videos, audios and multimedia files for web users. A user (client software) interacts with this data. A client can access (pull) data of responses from a server. The data can also publish (push) or post (after registering subscription) from a server. Internet applications including web sites, web services, web portals, online business applications, emails, chats, tweets and social networks provide and consume the web data.

Q.2 ***How big data add value to business?***

Ans. Big data analytics helps businesses to transform raw data into meaningful and actionable insights that can shape their business strategies. The most important contribution of big data to business is data-driven business decisions. Big data makes it possible for organizations to base their decisions on tangible information and insights.

Q.3 ***Why is big data analytics important?***

Ans. Most important advantage of big data analysis is, it helps organizations harness their data and use it to identify new opportunities.

Q.4 ***What are the sources of unstructured data in big data?***

Ans. The sources of unstructured data are as follows:

(i) Textfiles and documents
(ii) Server website and application log
(iii) Sensor data
(iv) Images, videos and audio files
(v) Emails
(vi) Social media data

Q.5 ***What is HDFS?***

Ans. HDFS : The HDFS is Hadoop's default storage unit and is responsible for storing different types of data in a distributed environment.

Q.6 ***What is big data?***

Ans. Big Data : Big data is a field that treats ways to analyz systematically extract information from, or otherwise de with data sets that are too large or complex to be dealt wi by traditional data-processing application software.

Q.7 ***List out the best practices of big data analytics.***

Ans. Best Practices of Big Data Analytics :

(i) Start at the end.
(ii) Build an analytical culture.
(iii) Re-engineer data systems for analytics.
(iv) Focus on useful data islands.
(v) Iterate often.

Velocity is also large. A number of satellites collect this data round the clock. Big Data analytics helps in drawing of maps of wind velocities, temperatures and other whether parameters.

Variety of images can be in visible range, such as IR-1 (infrared range -1), IR-2(infrared range -2), shortwave infrared (SWIR), MIR (medium range IR) and colour composite.

Data veracity, uncertain or imprecise data, is as important as volume, velocity and variety. Uncertainty arises due to poor resolutions used for recording or noise in images due to signal impairments.

Data processing needs increased speed of computations due to higher volumes. Need of data management, storage and increased analytics requires new innovative non-traditional methods.

Big Data of satellites helps in predicting weather, and mapping of different crops and from that estimating the expected crop yield.

Q.17 ***Write short note on Big Data types.***

Ans. A task team on Big Data classified the types of Big Data (June 2013). Another team from IBM developed a different classification for Big Data types.

Following are the suggested types:

1. Social networks and web data, such as Facebook, Twitter, e-mails, blogs and YouTube.
2. Transactions data and Business Processes (BPs) data, such as credit card transactions, flight bookings, etc. and public agencies data such as medical records, insurance business data etc.
3. Customer master data, such as data for facial recognition and for the name, date of birth, marriage anniversary, gender, location and income category.
4. Machine-generated data, such as machine-to-machine or Internet of Things data, and the data from sensors, trackers, web logs and computer systems log. Computer generated data is also considered as machine generated data from data store. Usage of programs for processing of data using data repositories, such as database or file, generates data and also machine generated data.
5. Human-generated data such as biometrics data, human-machine interaction data, e-mail records with a mail server and MySQL database of student grades. Humans also records their experiences in ways such as writing these in notebooks or diaries, taking photographs or audio and video clips. Human-source information is now almost entirely digitized and stored everywhere from personal computers to social networks. Such data are loosely structured and often ungoverned.

Q.18 ***Explain nature of data and its properties.***

Ans. Data : Data is a set of values of qualitative or quantitative variables; restated, pieces of data are individual pieces of information. Data is measured, collected and reported, and analyzed, where upon it can be visualized using graphs or images.

Properties of Data : For examining the properties of data reference to the various definitions of data. Reference to these definitions reveals that following are the properties of data:

(i) Amenability of use
(ii) Clarity
(iii) Accuracy
(iv) Essence
(v) Aggregation
(vi) Compression
(vii) Refinement

(i) Amenability of Use: From the dictionary meaning of data it is learnt that data are facts used in deciding something. In short, data are meant to be used as a base for arriving at definitive conclusions.

(ii) Clarity: Data are a crystallized presentation. Without clarity, the meaning desired to be communicated will remain hidden.

(iii) Accuracy: Data should be real, complete and accurate. Accuracy is thus, an essential property of data.

(iv) Essence: A large quantities of data are collected and they have to be compressed and refined. Data so refined can present the essence or derived qualitative value, of the matter.

(v) Aggregation: Aggregation is cumulating or adding up.

(vi) Compression: Large amounts of data are always compressed to make them more meaningful. Compre

data to a manageable size. Graphs and charts are some examples of compressed data.

(vii) **Refinement:** Data require processing or refinement. When refined, they are capable of leading to conclusions or even generalizations. Conclusions can be drawn only when data are processed or refined.

Q.19 ***Explain in detail about storage considerations in Big Data.***

Ans. In any environment intended to support the analysis of massive amounts of data, there must be the infrastructure supporting the data lifecycle from acquisition, preparation, integration, and execution. The need to acquire and manage massive amounts of data suggests a need for specialty storage systems to accommodate the big data applications. When evaluating specialty storage offerings, some variables to consider include:

(i) Scalability, which looks at whether expectations for performance improvement are aligned with the additional of storage resources, and the degree to which the storage subsystem can support massive data volumes of increasing size.

(ii) Extensibility, which examines how flexible the storage system's architecture is in allowing the system to be grown without the constraint of artificial limits.

(iii) Accessibility, which looks at any limitations or constraints in providing simultaneous access to an expanding user community without compromising performance.

(iv) Fault tolerance, which imbues the storage environment with the capability to recover from intermittent failures.

(v) High-speed I/O capacity, which measures whether the input/output channels can satisfy the demanding timing requirements for absorbing, storing, and sharing large data volumes.

(vi) Integrability, which measures how well the storage environment can be integrated into the production environment.

Often, the storage framework involves a software layer for managing a collection of storage resources and providing much of these capabilities. The software configures storage for replication to provide a level of fault tolerance, as well as managing communications using standard protocols (such as UDP or TCP/IP) among the different processing nodes. In addition, some frameworks will replicate stored data, providing redundancy in the event of a fault or failure.

Q.20 ***What are the various sources of Big Data?***

Ans. Sources of Big Data : Here various sources of big data are briefed. Digitization of content by industries is the new source of data (Villars et al., 2011). Advancements in technology also lead to high rate of data generation. For example, one of the biggest surveys in Astronomy, Sloan Digital Sky Survey (SDSS) has recorded a total of 25TB data during their first (2000-2005) and second surveys (2005-2008) combined. With the advancements in the resolution of the telescope, the amount of data collected at the end of their third survey (2008-14) is 100 TB. Use of "smart" instrumentation is another source of big data. Smart meters in the energy sector record the electricity utilization measurement every 15 minutes as compared to monthly readings before.

In addition to social media, Internet of Things (IoT) has, now, become the new source of data. The data can be captured from agriculture, industry, medical care, etc of the smart cities developed based on IoT. Table summarizes the various types of data produced in different sectors.

Table : Different Sources of Data

Sector	Data Produced	Use
Astronomy	Movement of stars, satellites, etc.	To monitor the activities of asteroid bodies and satellites
Financial	News content via video, audio, twitter and news report	To make trading decisions
Healthcare	Electronic medical records and images	To aid in short-term public health monitoring and long-term epidemiological research programs
Internet of Things (IoT)	Sensor data	To monitor various activities in smart cities

Life Sciences	Gene sequences	To analyze genetic variations and potential treatment effectiveness
Media/Entertainment	Content and user viewing behavior	To capture more viewers
Social Media	Blog posts, tweets, social networking sites, log details	To analyze the customer behavior pattern
Telecommunications	Call Detail Records (CDR)	Customer churn management
Transportation, Logistics, Retail, Utilities	Sensor data generated from fleet transceivers, RFID tag readers and smart meters	To optimize operations
Video Surveillance	Recordings from CCTV to IPTV cameras and recording system	To analyze behavioral patterns for service enhancement and security

Q.21 ***Write short note on Hadoop architecture.***

Ans. Hadoop framework includes following four modules:

(i) **Hadoop Common:** These are Java libraries and utilities required by other Hadoop modules. These libraries provides file system and OS level abstractions and contains the necessary Java files and scripts required to start Hadoop.

(ii) **Hadoop YARN:** This is a framework for job scheduling and cluster resource management.

(iii) **Hadoop Distributed File System (HDFS):** A distributed file system that provides high-throughput access to application data.

(iv) **Hadoop MapReduce:** This is YARN-based system for parallel processing of large data sets.

We can use following diagram to depict these four components available in Hadoop framework.

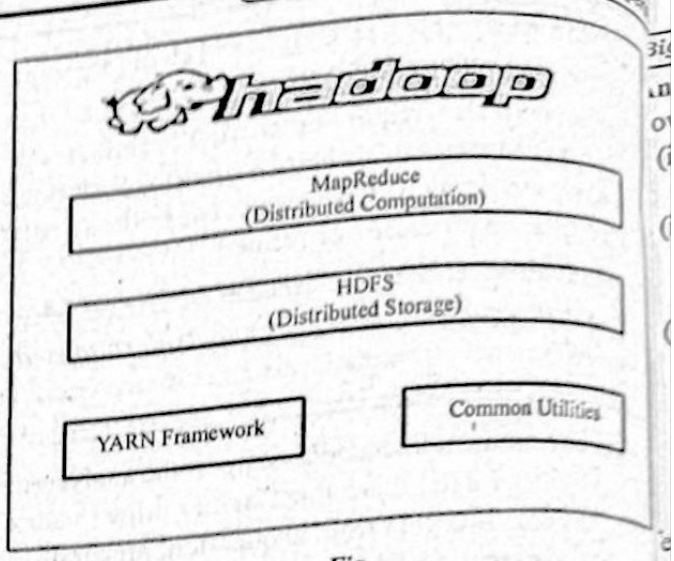


Fig.

Since 2012, the term "Hadoop" often refers not just the base modules mentioned above but also to the collection of additional software packages that can be installed on top of or alongside Hadoop, such as Apache Pig, Apache Hive, Apache HBase, Apache Spark etc.

Q.22 ***Explain the role of a JobTracker.***

Ans. The primary function of the JobTracker is resource management, which essentially means managing the TaskTrackers. Apart from this, JobTracker also tracks resource availability and handles task life cycle management (track the progress of tasks and their fault tolerance).

Some crucial features of the JobTracker are:

(i) It is a process that runs on a separate node (not on DataNode).

(ii) It communicates with the NameNode to identify data location.

(iii) It tracks the execution of MapReduce workloads.

(iv) It allocates TaskTracker nodes based on the available slots.

(v) It monitors each TaskTracker and submits the overall job report to the client.

(vi) It finds the best TaskTracker nodes to execute specific tasks on particular nodes.

Q.23 ***Can you recover a NameNode when it is down? If so, how?***

Ans. Yes, it is possible to recover a NameNode when it is down. Here's how you can do it:

(i) Use the FsImage (the file system metadata replica) to launch a new NameNode.

(ii) Configure DataNodes along with the clients so that they can acknowledge and refer to newly started NameNode.

(iii) When the newly created NameNode completes loading the last checkpoint of the FsImage (that has now received enough block reports from the DataNodes) loading process, it will be ready to start serving the client.

However, the recovery process of a NameNode is feasible only for smaller clusters. For large Hadoop clusters, the recovery process usually consumes a substantial amount of time, thereby making it quite a challenging task.

Q.24 ***Write advantages and disadvantages of large sized chunks in Google File System.***

Ans.

(i) It reduces clients need to interact with the master because reads and writes on the same chunk require only one initial request to the master for chunk location information.

(ii) Since on a large chunk, a client is more likely to perform many operations on a given chunk, it can reduce network overhead by keeping a persistent TCP connection to the chunk server over an extended period of time.

(iii) It reduces the size of the metadata stored on the master. This allows us to keep the metadata in memory, which in turn brings other advantages.

(iv) Lazy space allocation avoids wasting space due to internal fragmentation.

(v) Even with lazy space allocation, a small file consists of a small number of chunks, perhaps just one. The chunk servers storing those chunks may become hot spots if many clients are accessing the same file. In practice, hot spots have not been a major issue because the applications mostly read large multi-chunk files sequentially. To mitigate it, replication and allowance to read from other clients can be done.

Q.25 ***Explain Big Data Classification.***

Ans. Big Data Classification : Big Data can be classified on the basis of its characteristics that are used for designing data architecture for processing and analytics. Table gives various classification methods for data and Big Data.

Table : Various classification methods for data and Big Data

Basis of Classification	Examples
Data sources (traditional)	Data storage such as records, RDBMS, distributed databases, row-oriented In-memory data tables, column-oriented In-memory data tables, data warehouse, server, machine-generated data, human-sourced data, Business Process (BP) data, Business Intelligence (BI) data
Data formats (traditional)	Structured and semi-structured
Big Data sources	Data storage, distributed file system, Operational Data Store (ODS), data marts, data warehouse, NoSQL database (MongoDB, Cassandra), sensors data, audit trail of financial transactions, external data such as web, social media, weather data, health records
Big Data formats	Unstructured, semi-structured and multi-structured data
Data stores structure	Web, enterprise or cloud servers, data warehouse, row-oriented data for OLTP, column-oriented for OLAP, records, graph database, hashed entries for key/value pairs
Processing data rates	Batch, near-time, real-time, streaming
Processing Big Data rates	High volume, velocity, variety and veracity, batch, near real-time and streaming data processing
Analysis types	Batch, scheduled, near real-time datasets analytics
Big Data processing methods	Batch processing (for example, using Map Reduce, Hive or Pig), real-time processing (for example, using SparkStreaming, SparkSQL, Apache Drill)
Data analysis methods	Statistical analysis, predictive analysis, regression analysis, Mahout, machine learning algorithms, clustering algorithms, classifiers, text analysis, social network analysis, location-based analysis, diagnostic analysis, cognitive analysis
Data usage	Human, business process, knowledge discovery, enterprise applications, Data stores

BDA.8

PART-C

Q.26 *What is Big Data platform? Explain in detail, also write various features of big data platform.*

Ans. Big Data Platform :

(i) Big Data platform is integrated IT solution for Big Data management which combines several software systems, software tools and hardware to provide easy to use tools system to enterprises.

(ii) It is a single one-stop solution for all Big Data needs of an enterprise irrespective of size and data volume. Big Data Platform is enterprise class IT solution for developing, deploying and managing Big Data.

(iii) There are several open source and commercial Big Data Platform in the market with varied features which can be used in Big Data environment.

(iv) Big data platform is a type of IT solution that combines the features and capabilities of several big data application and utilities within a single solution.

(v) It is an enterprise class IT platform that enables organization in developing, deploying, operating and managing a big data infrastructure /environment.

(vi) Big data platform generally consists of big data storage, servers, database, big data management, business intelligence and other big data management utilities.

(vii) It also supports custom development, querying and integration with other systems.

(viii) The primary benefit behind a big data platform is to reduce the complexity of multiple vendors/ solutions into a one cohesive solution.

(ix) Big data platform are also delivered through cloud where the provider provides an all inclusive big data solutions and services.

Features of Big Data Platform : Here are most important features of any good Big Data Analytics platform:

(i) Big Data platform should be able to accommodate new platforms and tool based on the business requirement. Because business needs can change due to new technologies or due to change in business process.

(ii) It should support linear scale-out

(iii) It should have capability for rapid deployment

(iv) It should support variety of data format

(v) Platform should provide data analysis and report tools

(vi) It should provide real-time data analysis software

(vii) It should have tools for searching the data through large data sets

Big data is a term for data sets that are so large or complex that traditional data processing applications are inadequate. Challenges include :

(i) Analysis

(ii) Capture

(iii) Data Curation

(iv) Search

(v) Sharing

(vi) Storage

(vii) Transfer

(viii) Visualization

(ix) Querying

(x) Updating

Information Privacy :

(i) The term often refers simply to the use of predictive analytics or certain other advanced methods to extract value from data, and seldom to a particular size of data set.

(ii) Accuracy in big data may lead to more confident decision making, and better decisions can result in greater operational efficiency, cost reduction and reduced risk.

(iii) Big data usually includes data sets with sizes beyond the ability of commonly used.

(iv) Software tools to capture, curate, manage, and process data within a tolerable elapsed time.

(v) Big data "size" is a constantly moving target.

(vi) Big data requires a set of techniques and technologies with new forms of integration to reveal insights from datasets that are diverse, complex, and of a massive scale.

Q.27 *Explain the various Big Data platforms.*

Ans. List of Big Data Platforms

1. Hadoop

(i) Hadoop is open-source, Java based programming framework and server software which is used to save and analyze data with the help of 100s or even 1000s of commodity servers in a clustered environment.

(ii) Hadoop is designed to storage and process large datasets extremely fast and in fault tolerant way.

(iii) Hadoop uses HDFS (Hadoop File System) for storing data on cluster of commodity computers. If any server goes down it know how to replicate the data and there is no loss of data even in hardware failure.

(iv) Hadoop is Apache sponsored project and it consists of many software packages which runs on the top of the Apache Hadoop system.

(v) Top Hadoop based Commercial Big Data Analytics Platform.

(vi) Hadoop provides set of tools and software for making the backbone of the Big Data analytics system.

(vii) Hadoop ecosystem provides necessary tools and software for handling and analyzing Big Data.

(viii) On the top of the Hadoop system many applications can be developed and plugged-in to provide ideal solution for Big Data needs.

2. Cloudera

(i) Cloudera is one of the first commercial Hadoop based Big Data Analytics platform offering Big Data solution.

(ii) Its product range includes Cloudera Analytic DB, Cloudera Operational DB, Cloudera Data Science and Engineering and Cloudera Essentials.

(iii) All these products are based on the Apache Hadoop and provides real-time processing and analytics of massive data sets.

3. Amazon Web Services

(i) Amazon is offering Hadoop environment in cloud as part of its Amazon Web Services package.

(ii) AWS Hadoop solution is hosted solution which runs on Amazon's Elastic Cloud Compute and Simple Storage Service (S3).

(iii) Enterprises can use the Amazon AWS to run their Big Data processing analytics in the cloud environment.

(iv) Amazon EMR allows companies to setup and easily scale Apache Hadoop, Spark, HBase, Presto, Hive, and other Big Data Frameworks using its cloud hosting environment.

4. Hortonworks

(i) Hortonworks is using 100% open-source software without any propriety software. Hortonworks were the one who first integrated support for Apache HCatalog.

(ii) The Hortonworks is a Big Data company based in California.

(iii) This company is developing and supports application for Apache Hadoop.

Hortonworks Hadoop distribution is 100% open source and its enterprise ready with following features:

(i) Centralized management and configuration of clusters.

(ii) Security and data governance are built in feature of the system.

(iii) Centralized security administration across the system.

5. MapR

(i) MapR is another Big Data platform which us using the Unix file system for handling data.

(ii) It is not using HDFS and this system is easy to learn anyone familiar with the Unix system.

(iii) This solution integrates Hadoop, Spark, and Apache Drill with a real-time data processing feature.

6. IBM Open Platform

(i) IBM also offers Big Data Platform which is based on the Hadoop eco-system software.

(ii) IBM well knows company in software and data computing.

It uses the latest Hadoop software and provides following features (IBM Open Platform Features):

(i) Based on 100% open source software

(ii) Native support for rolling Hadoop upgrades

(iii) Support for long running applications within YEARN.

(iv) Support for heterogeneous storage which includes HDFS for in-memory and SSD in addition to HDD

(v) Native support for Spark, developers can use Java, Python and Scala to written program

(vi) Platform includes Ambari, which is a best tool for provisioning, managing and monitoring Apache Hadoop clusters

(vii) IBM Open Platform includes all the software of Hadoop ecosystem e.g. HDFS, YARN, MapReduce, Ambari, Hbase, Hive, Oozie, Parquet, Parquet Format, Pig, Snappy, Solr, Spark, Sqoop, Zookeeper, Open JDK, Knox, Slider

(viii) Developer can download the trial Docker Image or Native installer for testing and learning the system

(ix) Application is well supported by IBM technology team

7. Microsoft HDInsight

(i) The Microsoft HDInsight is also based on the Hadoop distribution and it's a commercial Big Data platform from Microsoft.

(ii) Microsoft is software giant which is into development of windows operating system for Desktop users and Server users.

(iii) This is the big Hadoop distribution offering which runs on the Windows and Azure environment.

(iv) It offers customized, optimized open source Hadoop based analytics clusters which uses Spark, Hive, MapReduce, HBase, Strom, Kafka and R Server which runs on the Hadoop system on windows/Azure environment.

Q.28 ***Explain types of data in detail.***

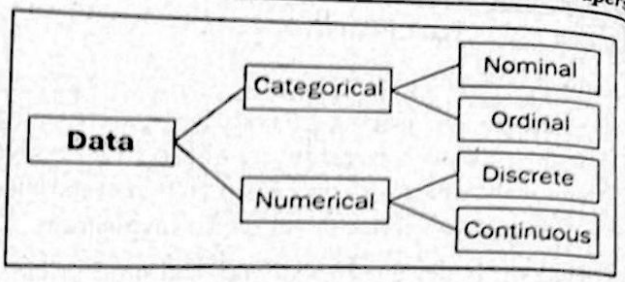
Ans. Types of Data

- In order to understand the nature of data it is necessary to categorize them into various types.
- Different categorizations of data are possible.
- The first such categorization may be on the basis of disciplines, e.g., Sciences, Social Sciences etc. in which they are generated.
- Within each of these fields, there may be several ways in which data can be categorized into types.

There are four types of data:

1. Nominal
2. Ordinal
3. Interval
4. Ratio

Each offers a unique set of characteristics, which impacts the type of analysis that can be performed.



The distinction between the four types of scales center on three different characteristics:

(i) The order of responses — whether it matters or not

(ii) The distance between observations — whether it matters or is interpretable

(iii) The presence or inclusion of a true zero

1. Nominal Scales : Nominal scales measure categories and have the following characteristics:

- **Order:** The order of the responses or observations does not matter.
- **Distance:** Nominal scales do not hold distance. The distance between a 1 and a 2 is not the same as a 2 and 3.
- **True Zero:** There is no true or real zero. In a nominal scale, zero is uninterruptible.
- **Appropriate Statistics for Nominal Scales:** Mode, count, frequencies.
- **Displays:** Histograms or bar charts.

2. Ordinal Scales : At the risk of providing a tautological definition, ordinal scales measure, well, order. So, our characteristics for ordinal scales are:

- **Order:** The order of the responses or observations matters.
- **Distance:** Ordinal scales do not hold distance. The distance between first and second is unknown as is the distance between first and third along with all observations.
- **True Zero:** There is no true or real zero. An item, observation, or category cannot finish zero.
- **Appropriate Statistics for Ordinal Scales:** Count, frequencies, mode.
- **Displays:** Histograms or bar charts.

3. Interval Scales : Interval scales provide insight into the variability of the observations or data. Classic interval scales are Likert scales (e.g., 1 - strongly agree and 9 - strongly

isagree) and Semantic Differential scales (e.g., 1 - dark nd 9 - light).

In an interval scale, users could respond to "I enjoy pening links to the website from a company email" with a esponse ranging on a scale of values. The characteristics of nterval scales are:

- **Order:** The order of the responses or observations does matter.
- **Distance:** Interval scales do offer distance. That is, the distance from 1 to 2 appears the same as 4 to 5. Also, six is twice as much as three and two is half of four. Hence, we can perform arithmetic operations on the data.
- **True Zero:** There is no zero with interval scales. However, data can be rescaled in a manner that contains zero. An interval scales measure from 1 to 9 remains the same as 11 to 19 because we added 10 to all values. Similarly, a 1 to 9 interval scale is the same a –4 to 4 scale because we subtracted 5 from all values. Although the new scale contains zero, zero remains uninterruptable because it only appears in the scale from the transformation.
- **Appropriate Statistics for Interval Scales:** Count, frequencies, mode, median, mean, standard deviation (and variance), skewness, and kurtosis.
- **Displays:** Histograms or bar charts, line charts, and scatter plots.

4. Ratio Scales : Ratio scales appear as nominal scales with a true zero.

They have the following characteristics:

- **Order:** The order of the responses or observations matters.
- **Distance:** Ratio scales do have an interpretable distance.
- **True Zero:** There is a true zero.

Income is a classic example of a ratio scale:

- Order is established. We would all prefer $100 to $1!
- Zero dollars means we have no income (or, in accounting terms, our revenue exactly equals our expenses!)
- Distance is interpretable, in that $20 appears as twice $10 and $50 is half of a $100.

For the web analyst, the statistics for ratio scales are the same as for interval scales.

- **Appropriate Statistics for Ratio Scales:** Count, frequencies, mode, median, mean, standard deviation (and variance), skewness, and kurtosis.
- **Displays:** Histograms or bar charts, line charts, and scatter plots.

Q.29 *Explain in detail about HDFS.*

Ans. Apache Hadoop is a collection of open-source software utilities that facilitate using a network of many computers to solve problems involving massive amounts of data and computation. It provides a software framework for distributed storage and processing of big data using the MapReduce programming model.

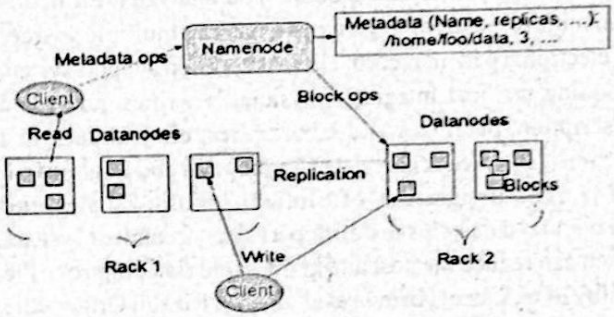


Fig. .

HDFS is a distributed file system that handles large data sets running on commodity hardware. It is used to scale a single Apache Hadoop cluster to hundreds (and even thousands) of nodes. HDFS is one of the major components of Apache Hadoop, the others being MapReduce and YARN. HDFS should not be confused with or replaced by Apache HBase, which is a column-oriented non-relational database management system that sits on top of HDFS and can better support real-time data needs with its in- memory processing engine.

Fast Recovery from Hardware Failures : Because one HDFS instance may consist of thousands of servers, failure of at least one server is inevitable. HDFS has been built to detect faults and automatically recover quickly.

Access to Streaming Data : HDFS is intended more for batch processing versus interactive use, so the emphasis in the design is for high data throughput rates, which accommodate streaming access to data sets.

Accommodation of Large Data Sets : HDFS accommodates applications that have data sets typically gigabytes to terabytes in size. HDFS provides high aggregate data bandwidth and can scale to hundreds of nodes in a single cluster.

Portability : To facilitate adoption, HDFS is designed to be portable across multiple hardware platforms and to be compatible with a variety of underlying operating systems.

Q.30 ***Write the various applications of Big Data analytics.***

Ans. Applications of Big Data Analytics : The concept of big data analytics has left no sector untouched. Few sectors like Telecommunication, Retail and Finance have been early adopters of big data analytics, followed by other sectors (Villars et al., 2011). The application of big data analytics in various sectors is discussed as follows:

Healthcare : Data analysts obtain and analyze information from multiple sources to gain insights. The multiple sources are electronic patient record; clinical decision support system including medical imaging, physician's written notes and prescription, pharmacy and laboratories; clinical data; and machine generated sensor data (Raghupathi and Raghupathi, 2014). The integration of clinical, public- health and behavioural data helps to develop a robust treatment system, which can reduce the cost and at the same time, improve the quality of treatment (Brown et al., 2011). Rizzoli Orthopedic Institute in Bologna, Italy analyzed the symptoms of individual patients to understand the clinical variations in a family. This helped to reduce the number of imaging and hospitalizations by 60% and 30%, respectively (Raghupathi and Raghupathi, 2014).

Obtaining information from external sources such as social media helps in early detection of epidemics and precautionary efforts. After the earthquake in Haiti in January 2010, analysis of tweets helped to track the spread of Cholera in the region (Raghupathi and Raghupathi, 2014). The data from the sensors are monitored and analyzed for adverse event prediction and safety monitoring (Mukherjee et al., 2012).

Artemis, a system developed by Blount et al. (2010), monitors and analyzes the physiological data from sensors in the intensive care units to detect the onset of medical complications, especially, in the case of neo-natal care. The real-time analysis of a huge number of claims requests can minimize fraud.

Telecommunication : Low adoption of mobile services and churn management are few of the most common problems faced by the mobile service providers (MSPs). The cost of acquiring new customer is higher than retaining the existing ones. Customer experience is correlated with customer loyalty and revenue (Soares, 2012a,b). In order to improve the customer experience, MSPs analyze a number of factors such as demographic data (gender, age, marital status, and language preferences), customer preferences, household structure and usage details (CDR, internet usage, value-added services (VAS)) to model the customer preferences and offer a relevant personalized service to them. This is known as targeted marketing, which improves the adoption of mobile services, reduces churn, thus, increasing the revenue of MSPs. Ufone, a Pakistan-based MSP, reduced the churn rate by precisely marketing the customized offers to their customers (Utsler, 2013). The company analyzes the CDR data to identify the call patterns to offer different plans to customers. The services are marketed to the customers through a call or text message. Their responses are recorded for further analysis.

Telecom companies are working towards combating telecom frauds. Often, traditional fraud management systems are poor at detecting new types of fraud. Even they detect the occurrence of fraud lately, by then fraudsters would have changed their strategy. In order to overcome the limitations of traditional fraud management system, MSPs are analyzing real-time data to minimize the losses due to fraud. Mobileum Inc., one of the leading telecom analytics solution providers, is working towards providing a real-time fraud detection system using predictive analytics and machine learning (Ray, 2015).

Network analytics is the next big thing in Telecom, where MSPs can monitor the network speed and manage the entire network. This helps to resolve the network problems within few minutes and helps to improve the quality of service and the customer experience. With the diffusion of Smartphones, based on analysis of real-time location and bevioural data, location-based services/context-based services can be offered to the customers when requested. This would increase the adoption of mobile services.

Financial Firms : Currently, capital firms are using advanced technology to store huge volumes of data. But increasing data sources like Internet and Social media require them to adopt big data storage systems. Capital markets are using big data in preparation for regulations like EMIR, Solvency II, Basel II etc, anti-money laundering, fraud mitigation, pre-trade decision-support analytics including sentiment analysis, predictive analytics and data tagging to identify trades (Verma

and Mani, 2012). The timeliness of finding value plays an important role in both investment banking and capital markets, hence, there is a need for real-time processing of data.

Retail : Evolution of e-commerce, online purchasing, social-network conversations and recently location-specific smartphone interactions contribute to the volume and the quality of data for data-driven customization in retailing (Brown et al., 2011). Major retail stores might place CCTV not only to observe the instances of theft but also to track the flow of customers (Villars et al., 2011). It helps to observe the age group, gender and purchasing patterns of the customers during weekdays and weekends. Based on the purchasing patterns of the customers, retailers group their items using a well-known data mining technique called Market Basket Analysis (proposed by (Agrawal and Srikant, 1994)), so that a customer buying bread and milk might purchase jam as well. This helps to decide on the placement of objects and decide on the prices (Brown et al., 2011; Villars et al., 2011). Nowadays, e-commerce firms use market basket analysis and recommender systems to segment and target the customers. They collect the click stream data, observe behavior and recommend products in the real time.

Analytics help the retail companies to manage their inventory. For example, Stage stores, one of the brand names of Stage Stores Inc. which operates in more 40 American states, used to analytics to forecast the order for different sizes of garments for different geographical regions (Meek, 2015).

Law Enforcement : Law enforcement officials try to predict the next crime location using past data i.e., type of crime, place and time; social media data; drone and smartphone tracking. Researchers at Rutgers University developed an app called RTM Dx to prevent crime and is being used by police department at Illinois, Texas, Arizona, New Jersey, Missouri and Colorado. With the help of the app, the police department could measure the spatial correlation between the location of crime and features of the environment (Mor, 2014).

A new technology called facial analytics that examines images of people without violating their privacy. Facial analytics is used to check child pomography. This saves the time of manual examination. Child pornography can be identified by integration of various technologies like Artemis and PhotoDNA by comparing files and image hashes with existing files to identify the subject as adult or child. It also identifies the cartoon based pornography (Ricanek and Boehnen, 2012).

Marketing : Marketing analytics helps the organizations to evaluate their marketing performance, to analyze the consumer behavior and their purchasing patterns, to analyze the marketing trends which would aid in modifying the marketing strategies like the positioning of advertisements in a webpage, implementation of dynamic pricing and offering personalized products (Soares, 2012a).

New Product Development : There is a huge risk associated with new product development. Enterprises can integrate both external sources, i.e., Twitter and Facebook page and internal data sources, i.e., customer relationship management (CRM) systems to understand the customers requirement for a new product, to gather ideas for new product and to understand the added feature included in a competitor's product. Proper analysis and planning during the development stage can minimize the risk associated with the product, increase the customer lifetime value and promote brand engagement (Anastasia, 2015). Ribbon UI in Microsoft 2007 was created by analyzing the customer data from previous releases of the product to identify the commonly used features and making intelligent decisions (Fisher et al., 2012).

Banking : The investment worthiness of the customers can be analyzed using demographic details, behavioral data, and financial employment. The concept of cross-selling can be used here to target specific customer segments based on past buying behavior, demographic details, sentiment analysis along with CRM data (Forsyth, 2012; Coumaros et al., 2014).

Energy and Utilities : Consumption of water, gas and electricity can be measured using smart meters at regular intervals of one hour. During this interval, a huge amount of data is generated and analyzed to change the patterns of power usage (Brown et al., 2011). The real-time analysis reveals energy consumption pattern, instances of electricity thefts and price fluctuations.

Insurance : Personalized insurance plan is tailored for each customer using updated profiles of changes in wealth, customer risk, home asset value, and other data inputs (Brown et al., 2011). Recently, driving data of customers such as miles driven, routes driven, time of day, and braking abruptness are collected by the insurance companies by using sensors in their cars. Comparing individual driving pattern and driver risk with the statistical information available such as peak hours of drivers on the road develops a personalized insurance plan. This analysis of driver risk and policy gives a competitive advantage to the insurance companies (Soares, 2012a; Sun and Heller, 2012).

Education : With the advent of computerized course modules, it is possible to assess the academic performance real time. This helps to monitor the performance of the students after each module and give immediate feedback on their learning pattern. It also helps the teachers to assess their teaching pedagogy and modify based on the students performance and needs. Dropout patterns, students requiring special attention and students who can handle challenging assignments can be predicted (West, 2012). Beck and Mostow (2008) studied the student reading comprehension using intelligent tutor software and observed that reading mistakes reduced considerably when the students re-read an old story instead of a new story.

Other Sectors : With increasing analytics skills among the various organizations, the advantage of big data analytics can be realized in sectors like construction and material sciences (Brown et al., 2011).

Q.31 ***Write the challenges of big data analytics in detail.***

Ans. Data is a very valuable asset in the world today. The economics of data is based on the idea that data value can be extracted through the use of analytics. Though Big data and analytics are still in their initial growth stage, their importance cannot be undervalued. As big data starts to expand and grow, the importance of big data analytics will continue to grow in everyday lives, both personal and business. In addition, the size and volume of data is increasing every single day, making it important to address the manner in which big data is addressed every day.

According to surveys being conducted many companies are opening up to using big data analytics in their daily functioning. With the rising popularity of Big data analytics, it is but obvious that investing in this medium is what is going to secure the future growth of companies and brands.

The key to data value creation is Big Data Analytics nd that is why it is important to focus on that aspect of nalytics. Many companies use different methods to employ ig Data analytics and there is no magic solution to cessfully implementing this. While data is important, even re, important is the process through which companies can insights with their help. Gaining insights from data is the of big data analytics and that is why investing in a system can deliver those insights is extremely crucial and rtant. Successful implementation of big data analytics, fore, requires a combination of skills, people and processes that can work in perfect synchronization with e other.

Today, companies are developing at a rapid pace, so are advancements in big technologies. This means brands must be ready to pilot and adopt big data in su manner that they become an integral aspect of the informa management and analytics infrastructure. With amaz potential, big data is today an emerging disruptive force is poised to become the next big thing in the field of integra analytics, thereby transforming the manner in which bra and companies perform their duties across stages economies.

With great potential and opportunities, however, co great challenges and hurdles. This means that companies m be able to solve all the concerned hurdles so that they unlock the full potential of big data analytics and its concern fields. When big data analytics challenges are addressed i proper manner, the success rate of implementing big da solutions automatically increases. As big data makes its w into companies and brands around the world, addressing the challenges is extremely important.

Some of the major challenges that big data analytics progra are facing today include the following :

1. Uncertainty of Data Management Landscape Because big data is continuously expanding, there are ne companies and technologies that are being developed ever day. A big challenge for companies is to find out whic technology works bests for them without the introduction o new risks and problems.

2. The Big Data Talent Gap: While Big Data is a growing field, there are very few experts available in this field. This i because Big data is a complex field and people who understand the complexity and intricate nature of this field are far few and between. Another major challenge in the field is the talent gap that exists in the industry.

3. Getting Data into the Big Data Platform: Data is increasing every single day. This means that companies have to tackle a limitless amount of data on a regular basis. The scale and variety of data that is available today can overwhelm any data practitioner and that is why it is important to make data accessibility simple and convenient for brand managers and owners.

4. Need for Synchronization across Data Sources: As data sets become more diverse, there is a need to incorporate them into an analytical platform. If this is ignored, it can create gaps and lead to wrong insights and messages.

5. Getting Important Insights through the Use of Big Data Analytics: It is important that companies gain proper insights from big data analytics and it is important that the correct department has access to this information. A major challenge in big data analytics is bridging this gap in an effective fashion.

Q.32 *What are the problems of traditional large scale systems with Big Data?*

Ans. Problem—Schema-On-Write: Traditional systems are schema-on-write. Schema-on-write requires the data to be validated when it is written. This means that a lot of work must be done before new data sources can be analyzed. Here is an example: Suppose a company wants to start analyzing a new source of data from unstructured or semi-structured sources. A company will usually spend months (3-6 months) designing schemas and so on to store the data in a data warehouse. That is 3 to 6 months that the company cannot use the data to make business decisions. Then when the data warehouse design is completed 6 months later, often the data has changed again. If you look at data structures from social media, they change on a regular basis. The schema-on-write environment is too slow and rigid to deal with the dynamics of semi-structured and unstructured data environments that are changing over a period of time. The other problem with unstructured data is that traditional systems usually use Large Object Byte (LOB) types to handle unstructured data, which is often very inconvenient and difficult to work with.

Solution—Schema-On-Read: Hadoop systems are schema-on-read, which means any data can be written to the storage system immediately. Data are not validated until they are read. This enables Hadoop systems to load any type of data and begin analyzing it quickly. Hadoop systems have extremely short business latency compared to traditional systems. Traditional systems require schema-on-write, which was designed more than 50 years ago. A lot of companies need real-time processing of data and customer models generated in hours or days versus weeks or months. The Internet of Things (IoT) is accelerating the data streams coming from different types of devices and physical objects, and digital personalization is accelerating the need to be able to make real-time decisions. Schema-on-read gives Hadoop a tremendous advantage over traditional systems in an area that matters most, that of being able to analyze the data faster to make business decisions. When working with complex data structures that are semi-structured or unstructured, schema-on- read enables data to be accessed much faster than schema-on-write systems.

Problem—Cost of Storage: Traditional systems use shared storage. As organizations start to ingest larger volumes of data, shared storage is cost prohibitive.

Solution—Local Storage: Hadoop can use the Hadoop Distributed File System (HDFS), a distributed file system that leverages local disks on commodity servers. Shared storage is about $1.20/GB, whereas local storage is about $0.04/GB. Hadoop's HDFS creates three replicas by default for high availability. So at 12 cents per GB, it is still a fraction of the cost of traditional shared storage.

Problem—Cost of Proprietary Hardware: Large proprietary hardware solutions can be cost prohibitive when deployed to process extremely large volumes of data. Organizations are spending millions of dollars in hardware and software licensing costs while supporting large data environments. Organizations are often growing their hardware in million dollar increments to handle the increasing data. New technology in traditional vendor systems that can grow to petabyte scale and good performance are extremely expensive.

Solution – Commodity Hardware: It is possible to build a high-performance super-computer environment using Hadoop. One customer was looking at a proprietary hardware vendor for a solution. The hardware vendor's solution was $1.2 million in hardware costs and $3 million in software licensing. The Hadoop solution for the same processing power was $400,000 for hardware, the software was free, and the support costs were included. Because data volumes would be constantly increasing, the proprietary solution would have grown in $500k and $1 million dollar increments, whereas the Hadoop solution would grow in $10,000 and $100,000 increments.

Problem—Complexity: When you look at any traditional proprietary solution. it is full of extremely complex silos of system administrators, DBAs, application server teams, storage teams, and network teams. Often there is one DBA for every 40 to 50 database servers. Anyone running traditional systems knows that complex systems fail in complex ways.

Solution—Simplicity: Because Hadoop uses commodity hardware and follows the "shared-nothing' architecture, it is a platform that one person can understand very easily. Numerous organizations running Hadoop have one administrator for every 1,000 data nodes. With commodity hardware, one person can understand the entire technology stack.

Problem—Causation: Because data is so expensive to store in traditional systems, data is filtered and aggregated, and large volumes are thrown out because of the cost of storage. Minimizing the data to be analyzed reduces the accuracy and confidence of the results. Not only are accuracy and confidence to the resulting data affected, but it also limits an organization's ability to identify business opportunities. Atomic data can yield more insights into the data than aggregated data.

Solution—Correlation: Because of the relatively low cost of storage of Hadoop, the detailed records are stored in Hadoop's storage system HDFS. Traditional data can then be analyzed with nontraditional data in Hadoop to find correlation points that can provide much higher accuracy of data analysis. We are moving to a world of correlation because the accuracy and confidence of the results are factors higher than traditional systems. Organizations are seeing big data as transformational. Companies building predictive models for their customers would spend weeks or months building new profiles. Now these same companies are building new profiles and models in a few days. One company would have a data load take 20 hours to complete, which is not ideal. They went to Hadoop and the time for the data load went from 20 hours to 3 hours.

Problem—Bringing Data to the Programs: In relational databases and data warehouses, data are loaded from shared storage elsewhere in the datacenter. The data must go over wires and through switches that have bandwidth limitations before programs can process the data. For many types of analytics that process 10s, 100s. and 1000s of terabytes, the capability of the computational side to process data greatly exceeds the storage bandwidth available.

Solution—Bringing Programs to the Data: With Hadoop, the programs are moved to where the data is. Hadoop data is spread across all the disks on the local servers that make up the Hadoop cluster, often in 64MB or 128MB block increments. Individual programs, one for every block, runs in parallel (up to the number of available map slots, more on this later) across the cluster, delivering a very high level of parallelization and Input/Output Operations per Second (IOPS). This means Hadoop systems can process extremely large volumes of data much faster than traditional systems and at a fraction of the cost because of the architecture model. Moving the programs (small component) to the data (large component) is an architecture that supports the extremely fast processing of large volumes of data.

Q.33 *Write detailed note on Hadoop.*

Ans. Hadoop : Doug Cutting, Mike Cafarella and team took the solution provided by Google and started an Open Source Project called HADOOP in 2005 and Doug named it after his son's toy elephant. Now Apache Hadoop is a registered trademark of the Apache Software Foundation. Hadoop runs applications using the MapReduce algorithm, where the data is processed in parallel on different CPU nodes. In short, Hadoop framework is capabale enough to develop applications capable of running on clusters of computers and they could perform complete statistical analysis for a huge amounts of data.

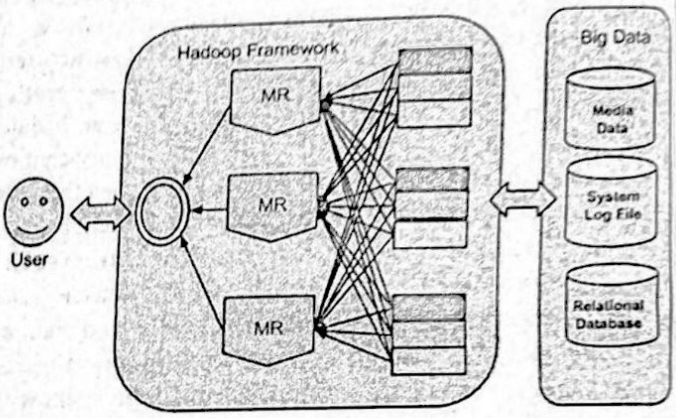


Fig.

Hadoop is an Apache open source framework written in java that allows distributed processing of large datasets across clusters of computers using simple programming models. A Hadoop frame-worked application works in an environment that provides distributed storage and computation across clusters of computers. Hadoop is designed to scale up from single server to thousands of machines, each offering local computation and storage.

Today, we're surrounded by data. People upload videos, take pictures on their cell phones, text friends, update their Facebook status, leave comments around the web, click on ads, and so forth. Machines, too, are generating and keeping more and more data. The exponential growth of data first presented challenges to cutting-edge businesses such as Google, Yahoo, Amazon, and Microsoft. They needed to go through terabytes and petabytes of data to figure out which websites were popular, what books were in demand, and what kinds of ads appealed to people. Existing tools were becoming inadequate to process such large data sets.

Google was the first to publicize MapReduce—a system they had used to scale their data processing needs. This system aroused a lot of interest because many other businesses were facing similar scaling challenges, and it wasn't feasible for everyone to reinvent their own proprietary tool.

Doug Cutting saw an opportunity and led the charge to develop an open source version of this MapReduce system called Hadoop. Soon after, Yahoo and others rallied around to support this effort. Today, Hadoop is a core part of the computing infrastructure for many web companies, such as Yahoo, Facebook, LinkedIn, and Twitter. Many more traditional businesses, such as media and telecom, are beginning to adopt this system too.

Hadoop, and large-scale distributed data processing in general, is rapidly becoming an important skill set for many programmers. An effective programmer, today, must have knowledge of relational databases, networking, and security, all of which were considered optional skills a couple decades ago. Similarly, basic understanding of distributed data processing will soon become an essential part of every programmer's toolbox. Leading universities, such as Stanford and CMU, have already started introducing Hadoop into their computer science curriculum.

Formally speaking, Hadoop is an open source framework for writing and running distributed applications that process large amounts of data. Distributed computing is a wide and varied field, but the key distinctions of Hadoop are that it is:

(i) **Accessible :** Hadoop runs on large clusters of commodity machines or on cloud computing services such as Amazon's Elastic Compute Cloud (EC2).

(ii) **Robust :** Because it is intended to run on commodity hardware, Hadoop is architected with the assumption of frequent hardware malfunctions. It can gracefully handle most such failures.

(iii) **Scalable :** Hadoop scales linearly to handle larger data by adding more nodes to the cluster.

(iv) **Simple :** Hadoop allows users to quickly write efficient parallel code.

Q.34 ***Write detailed note on GFS.***

Ans. GFS : Google File System is a scalable distributed file system (DFS) created by Google Inc. and developed to accommodate Google's expanding data processing requirements. GFS provides fault tolerance, reliability, scalability, availability and performance to large networks and connected nodes. GFS is made up of several storage systems built from low-cost commodity hardware components. It is optimized to accommodate Google's different data use and storage needs, such as its search engine, which generates huge amounts of data that must be stored.

The Google File System capitalized on the strength of off-the-shelf servers while minimizing hardware weaknesses. GFS is also known as GoogleFS.

GFS is enhanced for Google's core data storage and usage needs (primarily the search engine), which can generate enormous amounts of data that needs to be retained; Google File System grew out of an earlier Google effort, "Big Files", developed by Larry Page and Sergey Brin in the early days of Google, while it was still located in Stanford. Files are divided into fixed-size chunks of 64 megabytes, similar to clusters or sectors in regular file systems, which are only extremely rarely overwritten, or shrunk; files are usually appended to or read. It is also designed and optimized to run on Google's computing clusters, dense nodes which consist of cheap "commodity" computers, which means precautions must be taken against the high failure rate of individual nodes and the subsequent data loss.

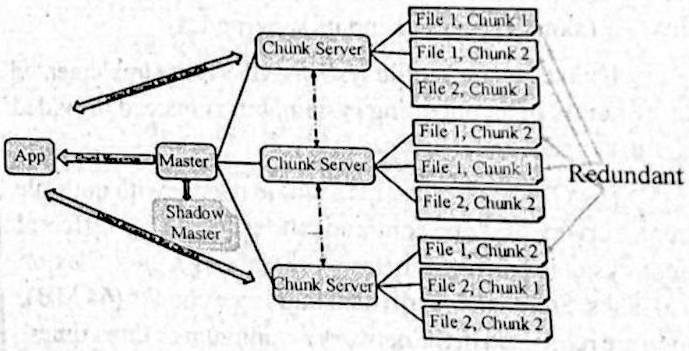


Fig.

A GFS cluster consists of multiple nodes. These nodes are divided into two types: One Master node and a large number of Chunk servers. Each file is divided into fixed-size chunks. Chunk servers store these chunks. Each chunk is assigned a unique 64-bit label by the master node at the time of creation, and logical mappings of files to constituent chunks are maintained. Each chunk is replicated several times throughout the network, with the minimum being three, but even more for files that have high end-in demand or need more redundancy.

The Master server does not usually store the actual chunks, but rather all the metadata associated with the chunks, such as the tables mapping the 64-bit labels to chunk locations and the files they make up, the locations of the copies of the chunks, what processes are reading or writing to a particular chunk, or taking a "snapshot" of the chunk pursuant to replicate it (usually at the instigation of the Master server, when, due to node failures, the number of copies of a chunk has fallen beneath the set number). All this metadata is kept current by the Master server periodically receiving updates from each chunk server ("Heart-beat messages").

Permissions for modifications are handled by a system of time-limited, expiring "leases", where the Master server grants permission to a process for a finite period of time during which no other process will be granted permission by the Master server to modify the chunk. The modifying chunk server, which is always the primary chunk holder, then propagates the changes to the chunk servers with the backup copies. The changes are not saved until all chunk servers acknowledge, thus guaranteeing the completion and atomicity of the operation.

Programs access the chunks by first querying the Master server for the locations of the desired chunks; if the chunks are not being operated on (i.e. no outstanding leases exist), the Master replies with the locations, and the program then contacts and receives the data from the chunk server directly (similar to Kazaa and its supernodes).

Unlike most other file systems, GFS is not implemented in the kernel of an operating system, but is instead provided as a user space library.

The GFS node cluster is a single master with multiple chunk servers that are continuously accessed by different client systems. Chunk servers store data as Linux files on local disks. Stored data is divided into large chunks (64 MB), which are replicated in the network a minimum of three times. The large chunk size reduces network overhead.

GFS is designed to accommodate Google's large cluster requirements without burdening applications. Files are stored n hierarchical directories identified by path names. Metadata-uch as namespace, access control data, and mapping nformation is controlled by the master, which interacts with nd monitors the status updates of each chunk server through ned heartbeat messages. GFS features include :

i) Fault tolerance

ii) Critical data replication

iii) Automatic and efficient data recovery

iv) High aggregate throughput

v) Reduced client and master interaction because of lar chunk server size

vi) Namespace management and locking

vii) High availability

The largest GFS clusters have more than 1,000 node with 300 TB disk storage capacity. This can be accessed b hundreds of clients on a continuous basis.

Q.35 ***What is HDFS? Write its features, goals an advantages.***

Ans. When a dataset outgrows the storage capacity of single physical machine, it becomes necessary to partition across a number of separate machines. File systems tha manage the storage across a network of machines are calle distributed file systems. Since they are network-based, a the complications of network programming kick in, thu making distributed filesystems more complex than regular dis file systems. For example, one of the biggest challenges i making the file system tolerate node failure without sufferin data loss. Hadoop comes with a distributed file system called HDFS, which stands for Hadoop Distributed File system.

Hadoop can work directly with any mountable distributed file system such as Local FS, HFTP FS, S3 FS, and others, but the most common file system used by Hadoop is the Hadoop Distributed File System (HDFS).

The Hadoop Distributed File System (HDFS) is based on the Google File System (GFS) and provides a distributed file system that is designed to run on large clusters (thousands of computers) of small computer machines in a reliable, fault-tolerant manner.

HDFS uses a master/slave architecture where master consists of a single NameNode that manages the file system metadata and one or more slave DataNodes that store the actual data.

A file in an HDFS namespace is split into several blocks and those blocks are stored in a set of DataNodes. The NameNode determines the mapping of blocks to the DataNodes. The DataNodes takes care of read and write operation with the file system. They also take care of block creation, deletion and replication based on instruction given by NameNode.

HDFS provides a shell like any other file system and a list of commands are available to interact with the file system.

Advantages of HDFS

(i) Hadoop framework allows the user to quickly write and test distributed systems. It is efficient, and it automatic distributes the data and work across the machines and in turn, utilizes the underlying parallelism of the CPU cores.

(ii) Hadoop does not rely on hardware to provide fault-tolerance and high availability (FTHA), rather Hadoop library itself has been designed to detect and handle failures at the application layer.

(iii) Servers can be added or removed from the cluster dynamically and Hadoop continues to operate without interruption.

(iv) Another big advantage of Hadoop is that apart from being open source, it is compatible on all the platforms since it is Java based.

Hadoop File System was developed using distributed file system design. It is run on commodity hardware. Unlike other distributed systems, HDFS is highly fault tolerant and designed using low-cost hardware.

HDFS holds very large amount of data and provides easier access. To store such huge data, the files are stored across multiple machines. These files are stored in redundant fashion to rescue the system from possible data losses in case of failure. HDFS also makes applications available to parallel processing.

Features of HDFS

(i) It is suitable for the distributed storage and processing.

(ii) Hadoop provides a command interface to interact with HDFS.

(iii) The built-in servers of NameNode and DataNode help users to easily check the status of cluster.

(iv) Streaming access to file system data.

(v) HDFS provides file permissions and authentication.

Goals of HDFS

(i) **Fault Detection and Recovery :** Since HDFS includes a large number of commodity hardware, failure of components is frequent. Therefore HDFS should have mechanisms for quick and automatic fault detection and recovery.

(ii) **Huge Datasets :** HDFS should have hundreds of nodes per cluster to manage the applications having huge datasets.

(iii) **Hardware at Data :** A requested task can be done efficiently, when the computation takes place near the data. Especially where huge datasets are involved, it reduces the network traffic and increases the throughput.

Q.36 *Explain building blocks of Hadoop in detail.*

Ans. HDFS is responsible for storing data on the cluster in Hadoop. Files in HDFS are split into blocks before they are stored on cluster of size 64MB or 128MB. On a fully configured cluster, "running Hadoop" means running a set of daemons, or resident programs, on the different servers in your network. Programs which reside permanently in memory are called "Resident Programs". Daemon is a thread in Java, which runs in background and mostly created by JVM for performing background task like Garbage collection. Each daemon runs separately in its own JVM. These daemons have specific roles; some exist only on one server, some exist across multiple servers. The daemons include :

1. NameNode
2. DataNode
3. Secondary NameNode
4. JobTracker
5. TaskTracker

The above daemons are called as "Building Blocks of Hadoop".

1. NameNode : Let's begin with arguably the most vital of the Hadoop daemons—the NameNode . Hadoop employs a master/slave architecture for both distributed storage and distributed computation. The distributed storage system is called the Hadoop File System , or HDFS. The NameNode is the master of HDFS that directs the slave DataNode daemons to perform the low-level I/O tasks. The NameNode is the bookkeeper of HDFS; it keeps track of how your files are broken down into file blocks, which nodes store those blocks, and the overall health of the distributed file system. The function of the NameNode is memory and I/O intensive. As such, the server hosting the NameNode typically doesn't store any user data or perform any computations for a MapReduce program to lower the workload on the machine. This means that the NameNode server doesn't double as a DataNode or a TaskTracker.

There is unfortunately a negative aspect to the importance of the NameNode— it's a single point of failure

of your Hadoop cluster. For any of the other daemons, if their host nodes fail for software or hardware reasons, the Hadoop cluster will likely continue to function smoothly or you can quickly restart it. Not so for the NameNode.

2. DataNode : Each slave machine in your cluster will host a DataNode daemon to perform the grunt (thankless and menial) work of the distributed filesystem – reading and writing HDFS blocks to actual files on the local filesystem. When you want to read or write a HDFS file, the file is broken into blocks and the NameNode will tell your client which DataNode each block resides in. Your client communicates directly with the DataNode daemons to process the local files corresponding to the blocks. Furthermore, a DataNode may communicate with other DataNodes to replicate its data blocks for redundancy.

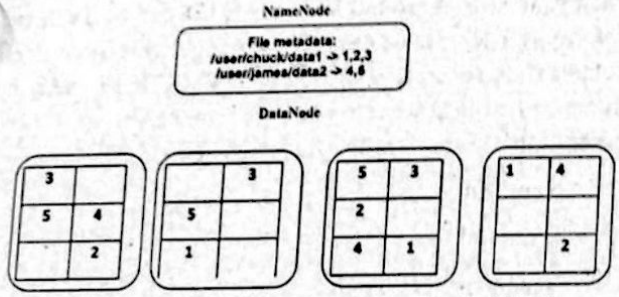


Fig. 1 : NameNode /DataNode interaction in HDFS

Fig.1 illustrates the roles of the NameNode and DataNodes. In this figure, we show two data files, one at / user/chuck/data 1 and another at /user/james/data 2. The data1 file takes up three blocks, which we denote 1, 2, and 3, and the data2 file consists of blocks 4 and 5. The content of the files are distributed among the DataNodes.

In this illustration, each block has three replicas. For example, block 1 (used for data1) is replicated over the three rightmost DataNodes. This ensures that if any one DataNode crashes or becomes inaccessible over the network, you'll still be able to read the files. DataNodes are constantly reporting to the NameNode. Upon initialization, each of the DataNodes informs the NameNode of the blocks it's currently storing. After this mapping is complete, the DataNodes continually poll the NameNode to provide information regarding local changes as well as receive instructions to create, move, or delete blocks from the local disk.

3. Secondary NameNode : The Secondary NameNode (SNN) is an assistant daemon for monitoring the state of the cluster HDFS. Like the NameNode, each cluster has one SNN, and it typically resides on its own machine as well. No other DataNode or TaskTracker daemons run on the same server. The SNN differs from the NameNode in that this process doesn't receive or record any real-time changes to HDFS. Instead, it communicates with the NameNode to take snapshots of the HDFS metadata at intervals defined by the cluster configuration. The NameNode is a single point of failure for a Hadoop cluster, and the SNN snapshots help minimize the downtime and loss of data. Nevertheless, a NameNode failure requires human intervention to reconfigure the cluster to use the SNN as the primary NameNode.

4. JobTracker : The JobTracker daemon is the liaison (communication/cooperation which facilitates a close working) between your application and Hadoop. Once you submit your code to your cluster, the JobTracker determines the execution plan by determining which files to process, assigns nodes to different tasks, and monitors all tasks as they're running. Should a task fail, the JobTracker will automatically relaunch the task, possibly on a different node up to a predefined limit of retries. There is only one JobTracker daemon per Hadoop cluster. It's typically run on a server as a master node of the cluster.

5. TaskTracker : As with the storage daemons, the computing daemons also follow a master/slave architecture: the JobTracker is the master overseeing the overall execution of a MapReduce job and the TaskTracker manage the execution of individual tasks on each slave node.

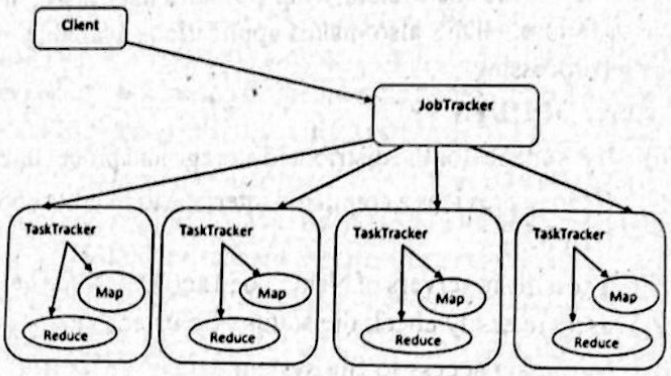


Fig. 2 : JobTracker and TaskTracker interaction

Fig.2 illustrates this interaction. Each TaskTracker is responsible for executing the individual tasks that the JobTracker assigns. Although there is a single TaskTracker per slave node, each TaskTracker can spawn multiple JVMs to handle many map or reduce tasks in parallel. One responsibility of the TaskTracker is to constantly communicate with the JobTracker. If the JobTracker fails to receive a heartbeat from a TaskTracker within a specified amount of time, it will assume the TaskTracker has crashed and will resubmit the corresponding tasks to other nodes in the cluster.

Having covered each of the Hadoop daemons, we depict the topology of one typical Hadoop cluster in the fig.3.

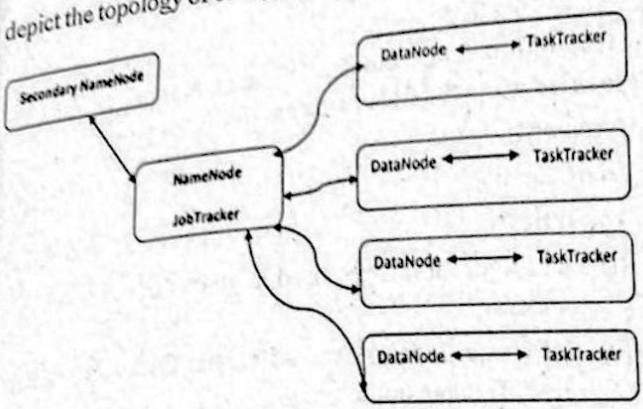


Fig. 3 : Topology of a typical Hadoop cluster

This topology features a master node running the NameNode and JobTracker daemons and a standalone node with the SNN in case the master node fails. For small clusters, the SNN can reside on one of the slave nodes. On the other hand, for large clusters, separate the NameNode and JobTracker on two machines. The slave machines each host a DataNode and TaskTracker, for running tasks on the same node where their data is stored.

Q.37 ***Explain process of configuring hadoop cluster and also describe various operational modes of Hadoop.***

Ans. When setting up a Hadoop cluster, you'll need to designate one specific node as the master node. Server will typically host the NameNode and JobTracker daemons. It'll also serve as the base station-contacting and activating the DataNode and TaskTracker daemons on all of the slave nodes. As such, we need to define a means for the master node to remotely access every node in your cluster.

Hadoop uses passphrase less SSH for this purpose. SSH utilizes standard public key cryptography to create a pair of keys for user verification—one public, one private. The public key is stored locally on every node in the cluster, and the master node sends the private key when attempting to access a remote machine. With both pieces of information, the target machine can validate the login attempt.

Define a Common Account : We've been speaking in general terms of one node accessing another; more precisely this access is from a user account on one node to another user account on the target machine. For Hadoop, the accounts should have the same username on all of the nodes and for security purpose we recommend it being a user-level account. This account is only for managing your Hadoop cluster. Once the cluster daemons are up and running, you'll be able to run your actual MapReduce jobs from other accounts.

Verify SSH Installation : The first step is to check whether SSH is installed on your nodes. We can easily do this by use of the "which" UNIX command:

```
[hadoop-user@master]$ which ssh
/usr/bin/ssh
[hadoop-user@master]$ which sshd
/usr/bin/sshd
[hadoop-user@master]$ which ssh-keygen
/ust/bin/ssh-keygen
```

If you instead receive an error message such as this,

```
/usr/bin/which: no ssh in (/usr/bin:/bin:/usr/sbin...
```

Install OpenSSH (www.openssh.com) via a Linux package manager or by downloading the source directly.

Operational Modes of Hadoop:

1. Local (Standalone) Mode : The standalone mode is the default mode for Hadoop. When you first uncompress the Hadoop source package, it's ignorant of your hardware setup. Hadoop chooses to be conservative and assumes a minimal configuration. All three XML files (or hadoopsite.xml before version 0.20) are empty under this default mode:

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl1"?>
<!-- Put site-specific property overrides in this file. -->
<configuration>
</configuration>
```

With empty configuration files, Hadoop will run completely on the local machine. Because there's no need to communicate with other nodes, the standalone mode doesn't use HDFS, nor will it launch any of the Hadoop daemons. Its primary use is for developing and debugging the application logic of a MapReduce program without the additional complexity of interacting with the daemons.

Properties:

(i) Default mode of Hadoop.

(ii) HDFS is not utilized in this mode.

(iii) Local File System is used for input and output.

(iv) Used for debugging purpose.

(v) Standalone is much faster than Pseudo-Distributed mode.

2. Pseudo-Distributed Mode : The pseudo-distributed mode is running Hadoop in a "cluster of one" with all daemons running on a single machine. This mode complements the standalone mode for debugging your code, allowing you to examine memory usage, HDFS input/output issues, and other daemon interactions. The below listing provides simple XML files to configure a single server in this mode.

core-site.xml

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<!-- Put site-specific property overrides in this file. -->
<configuration>
<property>
<name>fs.default .name</name>
<value>hdfs://localhost: 9000</value>
<description>The name of the default file system. A URI whose scheme and authority determine the FileSystem implementation.
</description>
</property>
</configuration>
```

mapred-site.xml

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<!-- Put site-specific property overrides in this file. -->
<configuration>
<property>
<name>mapred.job.tracker</name>
<value>localhost :9001</value>
<description>The host and port that the MapReduce job tracker runs
at.</description>
</property>
</configuration>
```

hdfs-site.xml

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<!-- Put site-specific property overrides in this file. -->
<configuration>
<property>
<name>dfs.replication</name>
<value>1</value>
<description>The actual number of replications can b
specified when the file is created.</description>
</property>
</configuration>
```

Properties:

(i) Configuration is required in given three files for thi node.

(ii) Here one node is used as Master/Data/Job Tracke Task Tracker node.

(iii) This is used for real code to test in HDFS.

(iv) This is a cluster, where all daemons are running o one node itself.

3. Fully-Distributed Mode : After continually emphasizing the benefits of distributed storage and distributed computation it's time for us to set up a full cluster. In the discussion below we'll use the following server names:

- **master**—The master node of the cluster and host o the NameNode and Job- Tracker daemons
- **backup**—The server that hosts the Secondary NameNode daemon
- hadoop1, hadoop2, hadoop3, ...—The slave boxes of the cluster running both DataNode and TaskTracker daemons

Using the preceding naming convention, the below listing is a modified version of the pseudo-distributed configuration files that can be used as a skeleton for your cluster's setup.

core-site.xml

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<!-- Put site-specific property overrides in this file. -->
<configuration>
<property>
<name>fs.default.name</name>
<value>hdfs://master:9000</value>
<description>The name of the default file system. A URI whose scheme and authority determine the FileSystem implementation.
</description>
</property>
</configuration>
```

mapred-site.xml

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<!-- Put site-specific property overrides in this file. -->
<configuration>
<property>
<name>mapred.job.tracker</name>
<value>master: 9001</value>
<description>The host and port that the MapReduce job tracker runs
at.</description>
</property>
</configuration>
```

hdfs-site.xml

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<!-- Put site-specific property overrides in this file. -->
<configuration>
<property>
<name>dfs.replication</name>
<value>3</value>
<description>The actual number of replications can be specified when the file is created.</description>
</property>
</configuration>
```

The key differences are :

(i) We explicitly stated the hostname for location of the NameNode and JobTracker daemons.

(ii) We increased the HDFS replication factor to take advantage of distributed storage. Recall that data is replicated across HDFS to increase availability and reliability.

Properties:

(i) This is a production phase.

(ii) Data are used and distributed across many nodes.

(iii) Different nodes will be used as Master node/Data node etc.

Q.38 *Explain how to configure xml files?*

Ans.

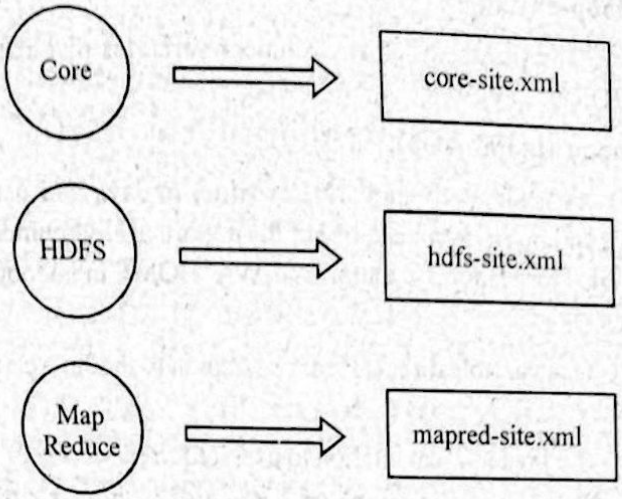


Fig.

Hadoop Cluster Configuration Files :

Configuration Filenames	Description of Log Files
Hadoop-env.sh	Environment variables that are used in the scripts to run Hadoop.
core-site.xml	Configuration settings for Hadoop Core such as I/O settings that are common to HDFS and MapReduce.
hdfs-site.xml	Configuration settings for HDFS daemons, the namenode, the secondary namenode and the data nodes.
mapred-site.xml	Configuration settings for MapReduce daemons; the job-tracker and the task-trackers.
masters	A list of machines (one per line) that each run a secondary namenode.
slaves	A list of machines (one per line) that each run a datanode and a task-tracker.

All these files are available under 'conf' directory of Hadoop installation directory.

Here is a listing of these files in the File System:

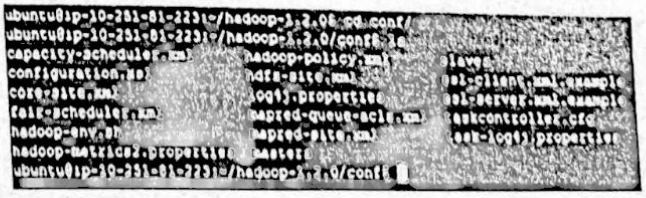


Fig.

Let's look at the files and their usage one by one!

adoop-env.sh

This file specifies environment variables that affect the JDK used by Hadoop.

Daemon (bin/hadoop).

As Hadoop framework is written in Java and uses Java Runtime environment, one of the important environment variables for Hadoop daemon is $JAVA_HOME in hadoop-env.sh.

This variable directs Hadoop daemon to the Java path in the system.

The java implementation to use. Required

export JAVA_HOME = /usr/lib/jvm/java-1.6.0-openjdk-amd64

This file is also used for setting another Hadoop daemon execution environment such as heap size (HADOOP_HEAP), hadoop home (HADOOP_HOME), log file location (HADOOP_LOG_DIR), etc.

Note: For the simplicity of understanding the cluster setup, we have configured only necessary parameters to start a cluster.

The following three files are the important configuration files for the runtime environment settings of a Hadoop cluster.

core-site.sh

This file informs Hadoop daemon where NameNode runs in the cluster. It contains the configuration settings for Hadoop Core such as I/O settings that are common to HDFS and MapReduce.

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xml" href= "configuration.xml"?>
<!-- Put site-specific property overrides in this file. -->
<configuration>
<property>
<name>fs.default.name</name>
<value>hdfs: //ec2-54-214-206-65.us-west-2.compute
amazonaws.com: 8020</value>
</property>
</configuration>
```

Where hostname and port are the machine and port on which NameNode daemon runs and listens. It also informs the Name Node as to which IP and port it should bind. The commonly used port is 8020 and you can also specify IP address rather than hostname.

hdfs-site.sh

This file contains the configuration settings for HDFS daemons; the Name Node, the Secondary Name Node, and the data nodes.

You can also configure hdfs-site.xml to specify default block replication and permission checking on HDFS. The actual number of replications can also be specified when the file is created. The default is used if replication is not specified in create time.

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xml" href="configuration.xml"?>
<i-- Put site-specific property overrides in this file. -->
<configuration>
<property>
<name>dfs.replication</name>
<value>3</value>
</property>
<property>
<name>dfs.permissions</name>
<value>false</value>
</property>
</configuration>
```

The value "true" for property 'dfs.permissions' enables permission checking in HDFS and the value "false" turns off the permission checking. Switching from one parameter value to the other does not change the mode, owner or group of files or directories.

mapred-site.sh

This file contains the configuration settings for MapReduce daemons; the job tracker and the task- trackers. Themapred.job.tracker parameter is a hostname (or IP address) and port pair on which the Job Tracker listens for RPC communication. This parameter specify the location of the Job Tracker to Task Trackers and MapReduce clients.

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xml" href="configuration.xml"?>
<i-- Put site-specific property overrides in this file. -->
<configuration>
<property>
```

<name>mapred.job.tracker</name>
<value>ec2-54-214-206-65.us-west-2. compute. amazonaws. com: 8021</value>
</property>
</configuration>

You can replicate all of the four files explained above to all the Data Nodes and Secondary Namenode. These files can then be configured for any node specific configuration e.g. in case of a different JAVA HOME on one of the Datanodes.

The following two file 'masters' and 'slaves' determine the master and slave Nodes in Hadoop cluster.

Masters : This file informs about the Secondary Namenode location to hadoop daemon. The 'masters' file at Master server contains a hostname Secondary Name Node servers.

Slaves : Contains a list of hosts, one per line, that are to host DataNode and TaskTracker servers.

Masters : Contains a list of hosts, one per line, that are to host Secondary NameNode servers.

The 'masters' file on Slave Nodes is blank.

Slaves : The 'slaves' file at Master node contains a list of hosts, one per line, that are to host Data Node and Task Tracker servers.

ec2-54-218-170-127 .us-west-2. compute. amazonaws.com
ec2-54-202-24-115 .us-west-2. compute. amazonaws.com

The 'slaves' file on Slave server contains the IP address of the slave node. Notice that the 'slaves' file at Slave node contains only its own IP address and not of any other Data Nodes in the cluster.

❑❑❑

Q.8 ***Write down the characteristics of big data applications.***

Ans. Characteristics of Big Data Applications :

(a) Data throttling
(b) Computation- restricted throttling
(c) Large data volumes
(d) Significant data variety
(e) Benefits from data parallelization

Q.9 ***Write down the four computing resources of big data storage.***

Ans. Computing Resources of Big Data Storage :

(i) Processing Capability
(ii) Memory
(iii) Storage
(iv) Network

Q.10 ***What are the three modes in which Hadoop can run?***

Ans. The three modes in which Hadoop can run are :

(i) Standalone Mode: This is the default mode. It uses the local File System and a single Java process to run the Hadoop services.

(ii) Pseudo-distributed Mode: This uses a single-node Hadoop deployment to execute all Hadoop services.

(iii) Fully-distributed Mode: This uses separate nodes to run Hadoop master and slave services.

Q.11 ***How can you restart NameNode and all the daemons in Hadoop?***

Ans. The following commands will help you restart NameNode and all the daemons:

You can stop the NameNode with ./sbin /Hadoop-daemon.sh stop NameNode command and then start the NameNode using ./sbin/Hadoop-daemon.sh start NameNode command.

You can stop all the daemons with ./sbin /stop-all.sh command and then start the daemons using the ./sbin/start-all.sh command.

Q.12 ***Define the Port Numbers for NameNode, Task Tracker and Job Tracker.***

Ans. NameNode — Port 50070
Task Tracker — Port 50060
Job Tracker — Port 50030

PART-B

Q.13 ***Explain the classification of data.***

Ans. Classification of Data : Data can be classified as structured, semi-structured, multi-structured and unstructured.

Structured data conform and associate with data schemas and data models. Structured data are found in tables (rows and columns). Nearly 15-20% data are in structured or semi-structured form. Unstructured data do not conform and associate with any data models.

Applications produce continuously increasing volumes of both unstructured and structured data. Data sources generate data in three forms, viz. structured, semi-structured and unstructured.

Using Structured Data : Structured data enables the following:

(i) Data insert, delete, update and append
(ii) Indexing to enable faster data retrieval
(iii) Scalability which enables increasing or decreasing capacities and data processing operations such as storing, processing and analytics
(iv) Transactions processing which follows ACID rules (Atomicity, Consistency, Isolation and Durability)
(v) Encryption and decryption for data security.

Using Semi-structured Data : Examples of semi-structured data are XML and JSON documents. Semi-structured data contain tags or other markers, which separate semantic elements and enforce hierarchies of records and fields within the data. Semi-structured form of data does not conform and associate with formal data model structures. Data do not associate data models, such as the relational database and table models.

Using Multi-structured Data : Multi-structured data refers to data consisting of multiple formats of data, viz. structured, semi-structured and/or unstructured data. Multi-structured data sets can have many formats. They are found in non-transactional systems. For example, streaming data on customer interactions, data of multiple sensors, data at web or enterprise server or the data- warehouse data in multiple formats.

Large-scale interconnected systems are thus required to aggregate the data and use the widely distributed resources efficiently.

Multi or semi-structured data has some semantic meanings and data is in both structured and unstructured formats. But as structured data, semi-structured data nowadays represent a few parts of data (5-10%). Semi-structured data type has a greater presence compared to structured data.

Using Unstructured Data : Unstructured data does not possess data features such as a table or a database. Unstructured data are found in file types such as .TXT, .CSV. Data may be as key-value pairs, such as hash key-value pairs. Data may have internal structures, such as in e-mails. The data do not reveal relationships, hierarchy relationships or object-oriented features, such as extendibility. The relationships, schema and features need to be separately established. Growth in data today can be characterised as mostly unstructured data.

Q.14 ***What are the characteristics of big data?***

Ans. Characteristics of big data, called 3Vs (and 4Vs also used) are:

(i) Volume : The phrase 'Big Data' contains the term big, which is related to size of the data and hence the characteristic. Size defines the amount or quantity of data, which is generated from applications. The size determines the processing considerations needed for handling that data.

(ii) Velocity : The term velocity refers to the speed of generation of data. Velocity is a measure of how fast the data generates and processes. To meet the demands and the challenges of processing big data, the velocity of generation of data plays a crucial role.

(iii)Variety : Big data comprises of a variety of data. Data is generated from multiple sources in a system. This introduces variety in data and therefore introduces 'complexity'. Data consists of various forms and formats. The variety is due to the availability of a large number of heterogeneous platforms in the industry. This means that the type to which big data belongs to is also an important characteristic that needs to be known for proper processing of data. This characteristic helps in effective use of data according to their formats, thus maintaining the importance of big data.

(iv) Veracity : It is also considered an important characteristic to take into account the quality of data captured, which can vary greatly, affecting its accurate analysis.

The 4Vs (i.e. volume, velocity, variety and veracity) data need tools for mining, discovering patterns, business intelligence, artificial intelligence (AI), machine learning (ML), text analytics, descriptive and predictive analytics, and the data visualization tools.

Q.15 ***How does such a toy company optimize the services offered, products and schedules, devise ways and use Big Data processing and storing for predictions using analytics?***

Ans. Assume that a retail and marketing company of toys uses several Big Data sources, such as (i) machine-generated data from sensors (RFID readers) at the toy packaging, (ii) transactions data of the sales stored as web data for automated reordering by the retail stores and (iii) tweets, Facebook posts, e-mails, messages, and web data for messages and reports.

The company uses Big Data for understanding the toys and themes in present days that are popularly demanded by children, predicting the future types and demands. The company using such predictive analytics, optimizes the product mix and manufacturing processes of toys. The company optimizes the services to retailers by maintaining toy supply schedules. The company sends messages to retailers and children using social media on the arrival of new and popular toys.

Q.16 ***Give an example of features of 3Vs in Big Data and application.***

Ans. Consider satellite images of the Earth's atmosphere and its regions. The volume of data from the satellites is large. A number of Indian satellites, such as KALPANA, INSAT-1A and INSAT-3D generate this data. Foreign satellites also generate voluminous data continuously. Satellites record the images of full disk and sectors, such as east and west Asia sectors and regions.