UNIT 1

Introduction to Big Data

CONTENTS

PART-1

Types of Digital Data, History of Big Data Innovation.

Questions-Answers

Long Answer Type and Medium Answer Type Questions

Que 1.1. Write short note on digital data.

Answer

1. Digital data is data that represents other forms of data using specific machine language systems that can be interpreted by various technologies.
2. One of the biggest strengths of digital data is that all sorts of complex analog input can be represented with the binary system.
3. For example, digital data is used is in cellphones or in MP3 players, digital thermometers and blood pressure meters as well as digital bathroom scales which give discrete but fast readings.
4. Numbers, text and other characters and symbols are naturally in a digital form.
5. Music, movies, and games can also be stored as what are ultimately sequences of 0's and 1's being interpreted by a computer.

Que 1.2. Explain different types of digital data.

Answer

Following are the different types of digital data :

1. **Structured digital data :**
   i. Structured digital data is data whose elements are addressable for effective analysis. It has been organized into a formatted repository that is typically a database.
   ii. It concerns all data which can be stored in database SQL in a table with rows and columns.

   iii. They have relational keys and can easily be mapped into pre-designed fields.
   iv. For example, Relational data.
2. **Semi-structured digital data :**
   i. Semi-structured digital data is information that does not reside in a relational database but that have some organizational properties that make it easier to analyze.
   ii. With some process, we can store them in the relation database, but semi-structured exist to ease space.
   iii. For example, XML data.
3. **Unstructured digital data :**
   i. Unstructured digital data is a data which is not organized in a predefined manner or does not have a predefined data model, thus it is not a good fit for a mainstream relational database.
   ii. So, for unstructured data, there are alternative platforms for storing and managing, it is increasingly prevalent in IT systems and is used by organizations in a variety of business intelligence and analytics applications.
   iii. For example, Word, PDF, Text, Media logs.

Que 1.3. Differentiate between structured, unstructured and semi-structured data.

Answer

S. No.	Properties	Structured digital data	Unstructured digital data	Semi-structured digital data
1.	Technology	It is based on Relational database table.	It is based on XML/RDF (Resource Description Framework).	It is based on character and binary data.
2.	Transaction management	Matured transaction and various concurrency techniques.	No transaction management and no concurrency.	Transaction is adapted from DBMS not matured.
3.	Version management	Versioning over tuples, row, tables.	Versioning over tuples or graph is possible.	Versioned as a whole.
4.	Flexibility	It is schema dependent and less flexible.	It is more flexible and there is absence of schema.	It is more flexible than structured data but less than unstructured data.
5.	Scalability	It is very difficult to scale DB schema.	It's scaling is simpler than structured data.	It is more scalable.
6.	Robustness	Very robust.	New technology, not very spread.	Less robust.
7.	Query performance	Structured query allow complex joining.	Queries over anonymous nodes are possible.	Only textual queries are possible.

Que 1.4. **Describe the history of Big data innovation.**

Answer

History of Big data is explained in three phases :

1. **Big data phase 1 :** During 1970 to 2000, it is a DBMS-based, structured content that include :
   i. RDBMS and data warehousing.
   ii. Extract transfer load.
   iii. Online analytical processing.
   iv. Dashboards and scorecards.
   v. Data mining and statistical analysis.
2. **Big data phase 2 :** During 2000 to 2010, it is a web-based unstructured content that include :
   i. Information retrieval and extraction.
   ii. Opinion mining.
   iii. Web analytics and web intelligence.
   iv. Social media analytics.
   v. Social network analysis.
   vi. Spatial-temporal analysis.
3. **Big data phase 3 :** During 2010 till today it is mobile and sensor-based content that include :
   i. Location-aware analysis.
   ii. Person-centered analysis.
   iii. Context-relevant analysis.
   iv. Mobile visualization.
   v. Human-computer interaction.

PART-2

Introduction to Big Data Platform, Drivers for Big Data.

Questions-Answers

Long Answer Type and Medium Answer Type Questions

Que 1.5. **Write a short note on : Big data platform.**

Answer

1. 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.
2. It is an enterprise class IT platform that enables organization in developing, deploying, operating and managing a big data infrastructure.
3. Big data platform consists of big data storage, servers, database, big data management, business intelligence and other big data management utilities.
4. It also supports custom development, querying and integration with other systems.
5. The primary benefit behind a big data platform is to reduce the complexity of multiple vendors/ solutions into a one cohesive solution.
6. Big data platform are also delivered through cloud where the provider provides an all inclusive big data solutions and services.

Que 1.6. Describe the drivers of Big data.

Answer

Following are the drivers of Big data :

1. **The digitization of society :**
   i. Big data is largely consumer driven and consumer oriented. Most of the data in the world is generated by consumers.
   ii. Most people consume and generate data through a variety of devices and (social) applications.
   iii. With every click, swipe or message, new data is created in a database somewhere around the world.
   iv. Because everyone now has a smartphone in their pocket, the data creation sums to incomprehensible amounts.
2. **The plummeting of technology costs :**
   i. Technology related to collecting and processing massive quantities of diverse (high variety) data has become increasingly more affordable.
   ii. The costs of data storage and processors keep declining, making it possible for small businesses and individuals to become involved with Big data.
3. **Connectivity through cloud computing :**
   i. Cloud computing environments have made it possible to quickly scale up or scale down IT infrastructure and facilitate a pay-as-you-go model.

   ii. This means that organizations that want to process massive quantities of data do not have to invest in large quantities of IT infrastructure.
4. **Increased knowledge about data science :**
   i. The knowledge and education about data science has greatly professionalized and more information becomes available every day.
   ii. While statistics and data analysis mostly remained an academic field previously, it is quickly becoming a popular subject among students and the working population.
5. **Social media applications :**
   i. Social media data provides insights into the behaviours, preferences and opinions of the public on a scale that have never been known before.
   ii. Due to this, it is immensely valuable to anyone who is able to derive meaning from these large quantities of data.
   iii. Social media data can be used to identify customer preferences for product development, target new customers for future purchases, or even target potential voters in elections.
6. **The upcoming Internet of Things (IoT) :**
   i. The Internet of things (IoT) is the network of physical devices, vehicles, home appliances and other items embedded with electronics, software, sensors, actuators, and network connectivity which enable these objects to connect and exchange data.
   ii. It is increasingly gaining popularity as consumer goods providers start including 'smart' sensors in household appliances.
   iii. Whereas the average household in 2015 had around 10 devices that connected to the internet, this number is expected to rise to 50 per household by 2025.

PART-3

Big Data Architecture and Characteristics, 5Vs of Big Data.

Questions-Answers

Long Answer Type and Medium Answer Type Questions

Que 1.7. Describe the architecture of Big data.

Answer

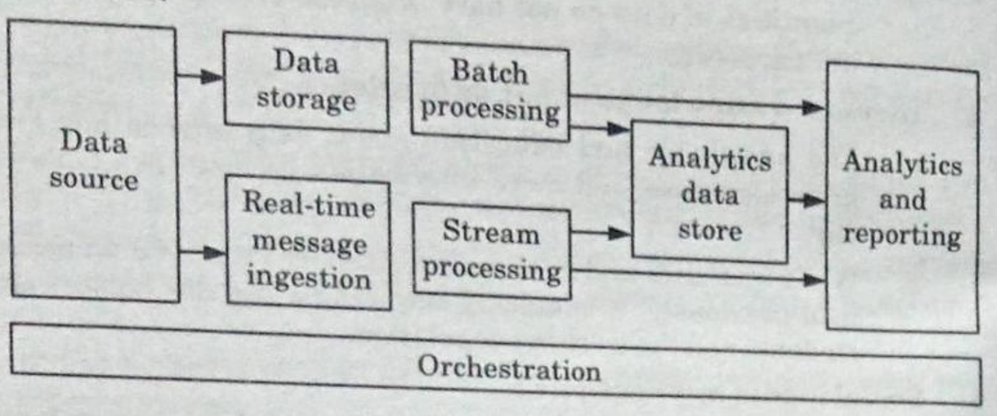


Fig. 1.7.1.

Most big data architectures include some or all of the following components :

1. **Data sources :**
   i. All big data solutions start with one or more data sources. Such as :
      a. Application data stores, relational databases.
      b. Static files produced by applications, such as web server log files.
      c. Real-time data sources, such as IoT devices.
2. **Data storage :**
   i. Data for batch processing operations is typically stored in a distributed file store that can hold high volumes of large files in various formats.
   ii. This kind of store is often called a data lake.
3. **Batch processing :**
   i. Because the data sets are so large, often a big data solution must process data files using long-running batch jobs to filter, aggregate, and otherwise prepare the data for analysis.
   ii. Usually these jobs involve reading source files, processing them, and writing the output to new files.
4. **Real-time message ingestion :**
   i. If the solution includes real-time sources, the architecture must include a way to capture and store real-time messages for stream processing.
   ii. This might be a simple data store, where incoming messages are dropped into a folder for processing.
   iii. However, many solutions need a message ingestion store to act as a buffer for messages, and to support scale-out processing, reliable delivery, and other message queuing semantics.

5. **Stream processing :**
   i. After capturing real-time messages, the solution must process them by filtering, aggregating, and otherwise preparing the data for analysis.
   ii. The processed stream data is then written to an output sink.
6. **Analytical data store :**
   i. Many big data solutions prepare data for analysis and then serve the processed data in a structured format that can be queried using analytical tools.
   ii. Data could be presented through a low-latency NoSQL technology such as HBase, or an interactive Hive database that provides a metadata abstraction over data files in the distributed data store.
7. **Analysis and reporting :**
   i. The goal of most big data solutions is to provide insights into the data through analysis and reporting.
   ii. To empower users to analyze the data, the architecture may include a data modeling layer, such as a multi-dimensional OLAP cube or tabular data model.
   iii. Analysis and reporting can also take the form of interactive data exploration by data scientists or data analysts.
8. **Orchestration :**
   i. Most big data solutions consist of repeated data processing operations, encapsulated in workflows that transform source data, move data between multiple sources and sinks, load the processed data into an analytical data store, or push the results straight to a report or dashboard.
   ii. To automate these workflows, we can use an orchestration technology.

Que 1.8. **What are the characteristics of Big data ?**

OR

Explain 5Vs of Big data.

OR

Discuss about the three dimensions of Big data.

Answer

Following are the characteristics/5Vs of Big data :

1. **Volume :**
   i. The name Big data itself is related to a size which is enormous. Size of data plays a very crucial role in determining value out of data.
   ii. Also, whether a particular data can actually be considered as a Big data or not, is dependent upon the volume of data.

iii. Hence, 'Volume' is one characteristic which needs to be considered while dealing with Big data.

2. **Variety :**
   i. Variety refers to heterogeneous sources and the nature of data, both structured and unstructured.
   ii. Data in the form of emails, photos, videos, monitoring devices, PDFs, audio, etc., are also being considered in the analysis applications.
   iii. This variety of unstructured data poses certain issues for storage, mining and analyzing data.

3. **Velocity :**
   i. The term 'velocity' refers to the speed of generation of data.
   ii. Big data velocity deals with the speed at which data flows in from sources like business processes, application logs, networks, and social media sites, sensors, mobile devices, etc.
   iii. The flow of data is massive and continuous.

4. **Value :**
   i. Value is the major issue that we need to concentrate on.
   ii. It is not just the amount of data that we store or process.
   iii. It is actually the amount of valuable, reliable and trustworthy data that needs to be stored, processed, analyzed to find insights.

5. **Veracity :**
   i. It refers to inconsistencies and uncertainty in data, that is, data which is available can sometimes get messy; quality and accuracy are difficult to control.
   ii. Big data is also variable because of the multitude of data dimensions resulting from multiple disparate data types and sources.
   iii. For example, Data in bulk could create confusion whereas less amount of data could convey half or incomplete information.

PART-4

Big Data Technology Component.

Questions-Answers

Long Answer Type and Medium Answer Type Questions

Que 1.9. What are the Big data technology components ?

Answer

Following are the components of Big data technology :

1. **Machine learning (ML) :**
   i. It is the science of making computers learns things by themselves.
   ii. In machine learning, a computer is expected to use algorithms and statistical models to perform specific tasks without any explicit instructions.
   iii. Machine learning applications provide results based on past experience.

2. **Natural Language Processing (NLP) :**
   i. It is the ability of a computer to understand human language as spoken.
   ii. For example, Google Home and Amazon Alexa, both use NLP and other technologies to give us a virtual assistant experience.

3. **Business Intelligence (BI) :**
   i. Business intelligence (BI) is a technology used for analysing data and delivering actionable information that helps executives, managers and workers make informed business decisions.
   ii. The ultimate goal of BI initiatives is to drive better business decisions that enable organizations to increase revenue and gain competitive advantages over business rivals.

4. **Cloud computing :**
   i. Cloud computing is the delivery of computing services including servers, storage, database, networking over the internet.
   ii. For example, Dropbox allow users to access files and store up to one terabyte of data.

PART-5

Big Data Importance and Application.

Questions-Answers

Long Answer Type and Medium Answer Type Questions

Que 1.10. Why Big data is important ?

Answer

Big data is important due to following reasons :

1. **Cost savings :**
   i. Tools of Big data like Hadoop and Cloud-Based Analytics can bring cost advantages to business when large amounts of data are to be stored.
   ii. These tools help in identifying more efficient ways of doing business.
2. **Time reductions :**
   i. The high speed of tools like Hadoop and in-memory analytics can easily identify new sources of data which helps businesses analyzing data immediately and make quick decisions based on the learning.
3. **Understand the market conditions :**
   i. By analyzing Big data we can get a better understanding of current market conditions.
4. **Control online reputation :**
   i. Big data tools can do sentiment analysis.
   ii. Therefore, we can get feedback about who is saying what about our company.
5. **Using Big data analytics to boost customer acquisition and retention :**
   i. The customer is the most important asset any business depends on.
   ii. There is no single business that can claim success without first having to establish a solid customer base.
   iii. Big data can be used for customer acquisition and retention.

Que 1.11. What are the applications of Big data ?

Answer

Following are the application of Big data :

1. **Health Care :** We have these days wearable devices and sensors that provide real-time updates to the health statement of a patient.
2. **Education :** A student's progress can be tracked and improved by proper analysis through big data analytics.
3. **Weather :**
   i. Weather sensors and satellites, which have been deployed around the globe collect data huge amounts and use that data to monitor the weather and environmental conditions.
   ii. Big data also helps to predict or forecast the weather conditions for the upcoming few days.

4. **Communication media and entertainment :**
   i. Big data helps in collecting, analyzing and utilizing consumer insight.
   ii. It also helps in understanding patterns of real-time, media content usage.
5. **Insurance :**
   i. Big data is used in industry to provide customer insights for transparent products, by analyzing and predicting customer behaviour through social media.
   ii. Big data also helps in better customer retention for insurance companies.

PART-6

Big Data Features, Security, Compliance, Auditing and Protection.

Questions-Answers
Long Answer Type and Medium Answer Type Questions

Que 1.12. What is Big data security ? What are the steps for securing Big data ?

Answer

1. The sheer size of a Big data repository brings with it a major security challenge.
2. Proper security entails more than just keeping the hackers out; it also means backing up data and protecting data from corruption.
3. Big data security is an umbrella term that includes all security measures and tools applied to analytics and data processes.
4. Attacks on big data systems can originate either from offline or online spheres and can crash a system.

Steps to securing Big data :

A. Get rid of unwanted data :

1. Securing the massive amounts of data can be addressed in several ways.
2. A starting point is to get rid of data that are no longer needed.
3. If you do not need certain information, it should be destroyed, because it represents a risk to the organization.

4. There are situations in which information cannot legally be destroyed; in that case, the information should be securely archived by an offline method.

B. Classifying data :

1. Protecting data becomes much easier if the data are classified.
2. The data should be divided into appropriate groupings for management purposes.
3. A classification system does not have to be very sophisticated or complicated to enable the security process.
4. Classification can become a powerful tool for determining the sensitivity of data.
5. Once organizations better understand their data, they can take important steps to segregate the information.

Que 1.13. Write a short note on : Big data and Compliance.

Answer

1. Compliance issues have a major effect on how big data is protected, stored, accessed, and archived.
2. Big data is not easily handled by the relational databases.
3. Big data is transforming the storage and access paradigms to an emerging world of horizontally scaling and unstructured databases.
4. This new world of file types and data is prompting analysis professionals to think of new problems to solve.
5. It is clear that a rebalancing of the database landscape is about to commence.
6. This has everything to do with compliance.
7. New data types and methodologies are still expected to meet the legislative requirements placed on businesses by compliance laws.
8. There will be no excuses accepted if a new data methodology breaks the law.
9. Preventing compliance from becoming the next big data nightmare is going to be the job of security professionals.

Que 1.14. Write a short note on : Protecting big data analytics.

Answer

1. Protecting data is an often forgotten inclination in the big data initiatives.
2. Big data contains all of the things you don't want to see when you are trying to protect data.

3. Big data can contain very unique sample sets that are accumulated frequently and in real time.
4. All of the data are unique to the moment, and if they are lost, they are impossible to recreate.
5. That uniqueness also means we cannot leverage time-saving backup preparation and security technologies.
6. This greatly increases the capacity requirements for backup subsystems, slows down security scanning, makes it harder to detect data corruption, and complicates archiving.
7. There is also the issue of the large size and number of files often found in Big data analytic environments.
8. Analytic information is often processed into an Oracle, NoSQL, or Hadoop environment, so real-time protection of that environment may be required.

PART-7

Big Data Privacy and Ethics.

Questions-Answers

Long Answer Type and Medium Answer Type Questions

Que 1.15. What is big data privacy ? Mention big data privacy concerns.

Answer

1. Big data privacy involves properly managing big data to minimize risk and protect sensitive data.
2. Because big data comprises large and complex data sets, many traditional privacy processes cannot handle the scale and velocity required.
3. To safeguard big data we need to create a framework for privacy protection that can handle the volume, velocity, variety, and value of big data.

Big data privacy concerns :

1. With more data spread across more locations, the risk of a privacy breach has never been higher.
2. Big data privacy is a matter of customer trust.
3. The more data you collect about users, the easier it gets to understand their current behavior, draw inferences about their future behavior,

and eventually develop deep and detailed profiles of their lives and preferences.

4. The more data you collect, the more important it is to be transparent with your customers.
5. The volume and velocity of data from existing sources is expanding fast.
6. To keep pace, your big data privacy strategy needs to expand, too.
7. That requires you to consider following issues :
   i. What do you intend to do with customer and user data ?
   ii. How accurate is the data, and what are the potential consequences of inaccuracies ?
   iii. How will your data security scale to keep up with threats of data breaches and insider threats ?
   iv. Where is your balancing point between the need to keep data locked down in-place and the need to expose it safely so you can extract value from it ?
   v. How do you maintain compliance with data privacy regulations that vary across the countries and regions where you do business ?
   vi. How do you maintain transparency about what you do with the big data you collect ?

Que 1.16. Explain principles of Big data ethics.

Answer

Following are the principles of Big data ethics :

1. **Private customer data and identity should remain private :** Privacy does not mean secrecy, as private data might need to be audited based on legal requirements, but that private data obtained from a person with their consent should not be exposed for use by other businesses or individuals with any traces to their identity.
2. **Shared private information should be treated confidentially :**
   i. Third party companies share sensitive data like medical, financial or locational and need to have restrictions on whether and how that information can be shared further.
   ii. Customers should have a transparent view of how their data is being used or sold, and the ability to manage the flow of their private information across third-party analytical systems.
3. **Big data should not interfere with human will :**
   i. Big data analytics can moderate and even determine who we are before we make up our own minds.
   ii. Companies need to begin to think about the kind of predictions and inferences that should be allowed and the ones that should not be.

4. Big data should not institutionalize unfair biases like racism or sexism. Machine learning algorithms can absorb unconscious biases in a population and amplify them via training samples.

PART-8

Big Data Analytics.

Questions-Answers

Long Answer Type and Medium Answer Type Questions

Que 1.17. Describe briefly Big data analytics.

Answer

1. The volume of data that one has to deal has exploded to unimaginable levels in the past decade, and at the same time, the price of data storage has systematically reduced.
2. Private companies and research institutions capture terabytes of data about their user's interactions, business, social media, and also sensors from devices such as mobile phones and automobiles.
3. Big data analytics involves collecting data from different sources, manage it in a way that it becomes available to be consumed by analysts and finally deliver data products useful to the organization business.
4. The process of converting large amounts of unstructured raw data, retrieved from different sources to a data product useful for organizations forms the core of Big data analytics.

Que 1.18. What are the advantages and disadvantages of Big data analytics ?

Answer

Advantages of Big data analytics :

1. It detects and corrects the errors from data sets using data cleansing.
2. Improves quality of data and give benefit to both customers and institution.
3. It removes duplicate information from data sets *i.e.*, save memory space.
4. It helps in displaying relevant advertisements on the online shopping websites based on historic data.

5. It helps in increasing revenue and productivity of the companies.
6. It reduces banking risks by identifying probable fraudulent customers based on historic data analysis.
7. It is used by security agencies for surveillance and monitoring purpose based on information collected by huge number of sensors.

Disadvantages of Big data analytics :

1. This may breach privacy of the customers as their information such as purchases, online transactions, subscriptions are visible to their parent companies.
2. The cost of data analytics tools vary based on applications and features supported.
3. The information obtained using data analytics can also be misused against group of people of certain country or community or caste.
4. It is very difficult to select the right data analytics tools.

PART-9

Challenges of Conventional System, Intelligent Data Analysis, Nature of Data.

Questions-Answers

Long Answer Type and Medium Answer Type Questions

Que 1.19. What is conventional system ? List some of the challenges of conventional systems.

Answer

1. The system consists of one or more zones each having either manually operated call points or automatic detection devices, or a combination of both.
2. Big data is huge amount of data which is beyond the processing capacity of conventional data base systems to manage and analyze the data in a specific time interval.

Challenges of conventional systems : Following are the challenges of conventional system :

1. **The uncertainty of data management landscape :**
   i. Because Big data is continuously expanding, there are new companies and technologies that are being developed every day.

   ii. A big challenge for companies is to find out which technology works bests for them without the introduction of new risks and problems.
2. **Talent gap in Big data :**
   i. While Big data is a growing field, there are very few experts available in this field.
   ii. This is because Big data is a complex field and people who understand the complexity and intricate nature of this field are few.
3. **Getting data into Big data structure :**
   i. Data is increasing every single day. This means that companies have to tackle limitless amount of data on a regular basis.
   ii. 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.
4. **Need for synchronization across data sources :**
   i. As data sets become more diverse, there is a need to incorporate them into an analytical platform.
   ii. 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 :**
   i. 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.
   ii. A major challenge in the Big data analytics is bridging this gap in an effective fashion.

Que 1.20. Explain intelligent data analysis.

Answer

1. Intelligent Data Analysis (IDA) reveals implicit, previously unknown and potentially valuable information or knowledge from large amounts of data.
2. Intelligent data analysis is also a kind of decision support process.
3. Based on artificial intelligence, machine learning, pattern recognition, statistics, database and visualization technology mainly, IDA automatically extracts useful information, necessary knowledge and interesting models from a lot of online data in order to help decision makers make the right choices.

4. The process of IDA generally consists of the following three stages :

   i. **Data preparation :** Data preparation involves selecting the required data from the relevant data source and integrating this into a data set to be used for data mining.

   ii. **Rule finding :** Rule finding is working out rules contained in the data set by means of certain methods or algorithms.

   iii. **Result validation :** Result validation requires examining these rules, and result explanation is giving intuitive, reasonable and understandable descriptions using logical reasoning.

Que 1.21. What is data ? List the properties of data. Describe the types of data.

Answer

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

Properties of data : Following are the properties of data :

1. **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.
2. **Clarity :** Data are a crystallized presentation. Without clarity, the meaning desired to be communicated will remain hidden.
3. **Accuracy :** Data should be real, complete and accurate. Accuracy is thus, an essential property of data.
4. **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.
5. **Aggregation :** Aggregation is cumulating or adding up.
6. **Compression :** Large amounts of data are always compressed to make them more meaningful. Compress data to a manageable size. Graphs and charts are some examples of compressed data.
7. **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.

Types of data : Following are the types of data :

1. **Categorical data :**
   i. These are values or observations that can be sorted into groups or categories.
   ii. There are two types of categorical values, nominal and ordinal.
   iii. A nominal variable has no intrinsic ordering to its categories.
   iv. For example, housing is a categorical variable having two categories (own and rent).
   v. An ordinal variable has an established ordering.
2. **Numerical data :**
   i. These are values or observations that can be measured.
   ii. There are two kinds of numerical values, discrete and continuous.
   iii. Discrete data are values or observations that can be counted and are distinct and separate.
   iv. For example, number of lines in a code.
   v. Continuous data are values or observations that may take on any value within a finite or infinite interval.

PART-10

Analytics Process and Tools.

Questions-Answers

Long Answer Type and Medium Answer Type Questions

Que 1.22. Explain the steps involved in analytic process.

Answer

Following are the steps of analytic process :

Step 1 : Deployment :

1. In this phase, we need to plan the deployment, monitoring and maintenance.
2. We produce a final report and review the project.
3. In this phase, we deploy the results of the analysis. This is also known as reviewing the project.

Step 2 : Business Understanding :

1. Business objectives are defined in this phase.
2. Whenever any requirement occurs, we need to assess the situation, determine data mining goals and then produce the project plan as per the requirement.

Step 3 : Data Exploration :

1. The step consists of data understanding.
2. This is necessary to verify the quality of data collected.
3. In this phase, we gather initial data, describe and explore the data and verify data quality to ensure it contains the data we require.
4. Data collected from the various sources is described in terms of its application and the need for the project in this phase. This is also known as data exploration.

Step 4 : Data Preparation :

1. From the data collected in the last step, we need to select data as per the need, clean it, construct it to get useful information and then integrate it all.
2. Finally, we need to format the data to get the appropriate data.
3. Data is selected, cleaned, and integrated into the format finalized for the analysis in this phase.

Step 5 : Data Modeling :

1. In this phase, we select a modeling technique, generate test design, build a model and assess the model built.
2. The data model is build to analyze relationships between various selected objects in the data.
3. Test cases are built for assessing the model and model is tested and implemented on the data in this phase.

Que 1.23. What are the tools used for analytic processes ?

Answer

A. Big data tools for high performance computing (HPC) and supercomputing :

1. Message Passing Interface (MPI)

B. Big data tools on clouds :

1. MapReduce model

2. Iterative MapReduce model
3. DAG model
4. Graph model
5. Collective model

C. Other BDA tools :

1. SaS
2. R
3. Hadoop

PART-11

Analytics Vs Reporting, Modern Data Analytics Tools.

Questions-Answers

Long Answer Type and Medium Answer Type Questions

Que 1.24. What is analysis ? What is reporting ? Differentiate between analysis and reporting.

Answer

Analysis :

1. Analysis is the process of exploring data and reports in order to extract meaningful insights, which can be used to better understand and improve business performance.
2. The goal of analysis is to answer questions by interpreting the data at a deeper level and providing actionable recommendations.
3. A firm may be focused on the general area of analytics (strategy, implementation, reporting, etc.), but not necessarily on the specific aspect of analysis.
4. Analysis transforms data and information into insights.

Reporting :

1. Reporting is the process of organizing data into informational summaries in order to monitor how different areas of a business are performing.
2. Measuring core metrics and presenting them falls under this category.

3. Reporting helps companies to monitor their online business and be alerted to when data falls outside of expected ranges.
4. Good reporting should raise questions about the business from its end users. Reporting translates raw data into information.

Difference between analysis and reporting : The basic differences between analysis and reporting are as follows :

S. No.	Analysis	Reporting
1.	Provides what is needed.	Provides what is asked for.
2.	Is typically customized.	Is typically standardized.
3.	Involves a person.	Does not involve a person.
4.	Is extremely flexible.	Is fairly inflexible.

Que 1.25. Describe modern data analytics tools.

Answer

Current modern analytic tools concentrate on following three classes :

A. Batch processing tools :

1. Batch processing system involves collecting a series of processing jobs and carrying them out periodically as a group (or batch) of jobs.
2. It allows a large volume of jobs to be processed at the same time.
3. One of the most famous and powerful batch process-based big data tool is Apache Hadoop.
4. It provides infrastructures and platforms for other specific big data applications.
5. Following are some batch processing tools :
   - i. **Apache Hadoop :** It is used to provide infrastructures and platforms for big data applications. It possesses high scalability, reliability and completeness.
   - ii. **Apache Mahout :** It is used to provide machine learning algorithms to businesses.
   - iii. **Talend Open Studio :** It is used to provide data management and application integration.

B. Stream processing tools :

1. Stream processing helps in predicting the life in data as and when it transpires.

2. The key strength of stream processing is that it can provide insights faster.
3. It helps understanding the hidden patterns in millions of data records in real time.
4. It processes the data from single or multiple sources in real or near-real time applying the desired business logic and emitting the processed information.
5. Following are some of the real time data streaming tools :
   - i. **Apache Storm :** It is a distributed real-time computation system. Its applications are designed as directed acyclic graphs. Storm is a stream processing engine without batch support.
   - ii. **Apache Flink :** It is a streaming data flow engine that provides communication fault tolerance and data distribution computation over data stream. It can execute both stream processing and batch processing easily. It is designed as an alternative to Map Reduce.
   - iii. **Amazon Kinesis :** It is an out of the box streaming data tool. It comprises of shards which Kafka calls partitions. It solves a variety of streaming data problems.

C. Interactive analysis tools :

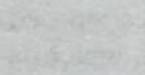
1. The interactive analysis presents the data in an interactive environment, allowing users to undertake their own analysis of information.
2. Users are directly connected to the computer and hence can interact with it in real time.
3. The data can be reviewed, compared and analyzed in tabular or graphic format or both at the same time.
4. Following are some interactive analysis tools :
   - i. **Google's Dremel :**
     - a. In 2010, Google proposed an interactive analysis system, named Dremel, which is scalable for processing nested data.
     - b. Dremel provides a very fast SQL like interface to the data by using a different technique than Map Reduce.
     - c. Dremel has a very different architecture compared with well-known Apache Hadoop.
     - d. Dremel has capability to run aggregation queries over trillion-row tables in seconds by means of combining multi-level execution trees and columnar data layout.

ii. Apache Drill :

a. It is an Apache open-source SQL query engine for big data exploration.

b. Drill is designed to support high-performance analysis on the semi-structured and rapidly evolving data coming from modern big data applications.

c. Drill provides plug-and-play integration with existing Apache Hive and Apache HBase deployments.



2 UNIT

Hadoop

CONTENTS

PART-1

History of Hadoop, Apache Hadoop.

Questions-Answers
Long Answer Type and Medium Answer Type Questions

Que 2.1. Describe the history of Hadoop.

Answer

1. In 2002, Doug Cutting and Mike Cafarella started to work on a project, Apache Nutch. It is an open source web crawler software project.
2. While working on Apache Nutch, they were dealing with big data. To store that data they have to spend a lot of costs which becomes the consequence of that project. This problem becomes one of the important reasons for the emergence of Hadoop.
3. In 2003, Google introduced a file system known as GFS (Google file system). It is a proprietary distributed file system developed to provide efficient access to data.
4. In 2004, Google released a white paper on Map Reduce. This technique simplifies the data processing on large clusters.
5. In 2005, Doug Cutting and Mike Cafarella introduced a new file system known as NDFS (Nutch Distributed File System). This file system also includes Map Reduce.
6. In 2006, Doug Cutting quit Google and joined Yahoo. On the basis of the Nutch project, Dough Cutting introduces a new project Hadoop with a file system known as HDFS (Hadoop Distributed File System). Hadoop first version 0.1.0 released in this year.
7. In 2007, Yahoo runs two clusters of 1000 machines.
8. In 2008, Hadoop became the fastest system to sort 1 terabyte of data on a 900 node cluster within 209 seconds.
9. In 2013, Hadoop 2.2 was released.
10. In 2017, Hadoop 3.0 was released.

Que 2.2. Write short note on Apache Hadoop.

Answer

1. Apache Hadoop is an open source framework that is used to efficiently store and process large datasets ranging in size from gigabytes to petabytes of data.

2. Hadoop allows clustering multiple computers to analyze massive datasets in parallel more quickly.
3. The Apache Hadoop software library is a framework that allows for the distributed processing of large data sets across clusters of computers using simple programming models.
4. It is designed to scale up from single servers to thousands of machines, each offering local computation and storage.
5. The Apache Hadoop framework is composed of the following modules :
   i. **Hadoop Common :** It contains libraries and utilities needed by other Hadoop modules.
   ii. **Hadoop Distributed File System (HDFS) :** A distributed file-system that stores data on the commodity machines, providing very high aggregate bandwidth across the cluster.
   iii. **Hadoop YARN :** A resource-management platform responsible for managing compute resources in clusters and using them for scheduling of users applications.
   iv. **Hadoop Map Reduce :** A programming model for large scale data processing.

PART-2

The Hadoop Distributed File System, Components of Hadoop, Data Format.

Questions-Answers
Long Answer Type and Medium Answer Type Questions

Que 2.3. What is Hadoop Distributed File System (HDFS) ? How does HDFS work ? Also explain the features of HDFS.

Answer

1. 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 commodity hardware.
2. The Hadoop Distributed File System (HDFS) is the primary data storage system used by Hadoop applications.
3. HDFS employs a NameNode and DataNode architecture to implement a distributed file system.
4. It is highly fault-tolerant and is designed to be deployed on low-cost hardware.

5. It provides high throughput access to application data and is suitable for applications having large datasets.

Working of HDFS :

1. The way HDFS works is by having a main NameNode and multiple DataNode on a commodity hardware cluster.
2. All the nodes are usually organized within the same physical rack in the data center.
3. Data is then broken down into separate blocks that are distributed among the various DataNodes for storage.
4. NameNode is the master daemon in HDFS. It runs on the master nodes.
5. It maintains the filesystem namespace.
6. NameNode does not store the actual data.
7. It stores the metadata, such as information about blocks of files, files permission, blocks locations, etc.
8. NameNode manages the DataNode and provides instructions to them.
9. DataNode is the slave daemon in HDFS.
10. DataNodes are the slave nodes that store the actual business data.
11. They are responsible for serving the client's read/write requests based on the instructions from NameNode.

Features of HDFS : Following are the features of HDFS :

1. **Data replication :** This is used to ensure that the data is always available and prevents data loss.
2. **Fault tolerance and reliability :** HDFS ability to replicate file blocks and store them across nodes in a large cluster ensures fault tolerance and reliability.
3. **High availability :** Due to replication across nodes the data is available even if the NameNode or a DataNode fails.
4. **Scalability :** Because HDFS stores data on various nodes in the cluster, as requirements increase, a cluster can scale to hundreds of nodes.
5. **High throughput :** Since HDFS stores data in a distributed manner, the data can be processed in parallel on a cluster of nodes, this cuts the processing time and enable high throughput.

Que 2.4. Describe the goals of HDFS.

Answer

Following are the goals of HDFS :

1. **Fault detection and recovery :**
   i. Since HDFS includes a large number of commodity hardware, failure of components is frequent.

   ii. Therefore, HDFS should have mechanisms for quick and automatic fault detection and recovery.
2. **Huge datasets :**
   i. HDFS should have hundreds of nodes per cluster to manage the applications having huge datasets.
3. **Hardware at data :**
   i. A requested task can be done efficiently, when the computation takes place near the data.
   ii. Especially where huge datasets are involved, it reduces the network traffic and increases the throughput.
4. **Handling the hardware failure :**
   i. The HDFS contains a multiple server machines.
   ii. If any machine fails, the HDFS goal is to recover it quickly.
5. **Streaming data access :**
   i. The HDFS applications usually run on the general purpose file system.
   ii. This application requires streaming access to their sets.
6. **Coherence model :**
   i. The application that runs on HDFS require to follow the write-once-read-many approach.
   ii. So, a file once generated need not to be changed. However, it can be appended and truncate.

Que 2.5. What are the benefits of using HDFS ?

Answer

Following are the main advantages of using HDFS :

1. **Cost effectiveness :** The DataNodes that store the data rely on inexpensive off-the-shelf hardware, which cuts storage costs. Also, because HDFS is open source, there's no licensing fee.
2. **Large data set storage :** HDFS stores a variety of data of any size – from megabytes to petabytes – and in any format, including structured and unstructured data.
3. **Fast recovery from hardware failure :** HDFS is designed to detect faults and automatically recover on its own.
4. **Portability :** HDFS is portable across all hardware platforms, and it is compatible with several operating systems, including Windows, Linux and Mac OS/X.
5. **Streaming data access :** HDFS is built for high data throughput, which is best for access to streaming data.

Que 2.6. What are various components of the Hadoop ?

Answer

Different components of the Hadoop are as follows :

1. **HDFS (Hadoop Distributed File System) :**
   i. It is the storage component of Hadoop that stores data in the form of files.
   ii. Each file is divided into blocks of 128MB (configurable) and stores them on different machines in the cluster.
   iii. It has a master-slave architecture with two main components : Name Node and Data Node.
2. **MapReduce :**
   i. To handle Big Data, Hadoop relies on the MapReduce algorithm.
   ii. It essentially divides a single task into multiple tasks and processes them on different machines.
   iii. It has two important phases : Map and Reduce.
   iv. Map phase filters, groups, and sorts the data.
   v Reduce phase aggregates the data, summarises the result, and stores it on HDFS.
3. **YARN :**
   i. YARN or Yet Another Resource Negotiator manages resources in the cluster and manages the applications over Hadoop.
   ii. It allows data stored in HDFS to be processed and run by various data processing engines.
4. **Hadoop Common :**
   i. It contains libraries and utilities needed by other Hadoop modules.

Que 2.7. What are the various data formats used in Hadoop ?

Answer

Following are the various data formats used in Hadoop :

1. **Text/CSV :**
   i. A plain text file (CSV) is the most common format both outside and within the Hadoop ecosystem.
   ii. It does not support block compression, so the compression of a CSV file in Hadoop can have a high cost in reading.
2. **SequenceFile :**
   i. The SequenceFile format stores the data in binary format.
   ii. This format accepts compression.

   iii. It does not store metadata and the only option in the evolution of its scheme is to add new fields at the end.
   iv. This is usually used to store intermediate data in the input and output of Map Reduce processes.
3. **Avro :**
   i. Avro is a row-based storage format.
   ii. This format includes in each file, the definition of the scheme of our data in JSON format, improving interoperability and allowing the evolution of the scheme.
   iii. Avro also allows block compression in addition to its divisibility, making it a good choice for most cases when using Hadoop.
4. **Parquet :**
   i. Parquet is a column-based binary storage format that can store nested data structures.
   ii. This format is very efficient in terms of disk input / output operations when the necessary columns to be used are specified.
   iii. This format is very optimized for use with Cloudera Impala.
5. **RCFile (Record Columnar File) :**
   i. RCFile is a columnar format that divides data into groups of rows, and inside it, data is stored in columns.
   ii. This format does not support the evaluation of the scheme and if we want to add a new column it is necessary to rewrite the file, which slows down the process.

PART-3

Analyzing Data with Hadoop.

Questions-Answers

Long Answer Type and Medium Answer Type Questions

Que 2.8. Give reasons why Hadoop can be considered a helpful tool to analyze the big data ?

Answer

Following reasons why Hadoop can be considered a helpful tool to analyze the big data :

A. Storage :

1. Storing data is one of the biggest challenges for traditional methods of handling large sets of incoming data.
2. HDFS is one of the components of Hadoop that helps maintaining the storage of big data.
3. HDFS consist of single cluster or multiple clusters. Every cluster consists of blocks which is 128 MB by default.
4. When a user defines input, the contents of the input are equally divided into the blocks.
5. And the data are replicated into the data nodes.
6. So, HDFS allows user to store the lots of dataset and when needed, more servers can be added at a very low cost.

B. Processing :

1. When the size of the dataset is larger, the time taken to process it is also longer while using the traditional methods.
2. More servers are added to store the large quantity of data, but server does not support the parallel computing.
3. In case of Hadoop the processing of data is done using parallel computing which saves the processing time.

C. Cost efficiency :

1. Maintaining the database at a minimum cost is the one of the most important challenges of the big data.
2. Companies using traditional method of handling big data are spending $25,000 to $50,000 per year for 1 terabyte of data.
3. Hadoop software can reduce this cost into few thousand dollars per terabyte per year.

D. Allows more data to capture :

1. Due to cost related issues many companies do not capture the large volume of data.
2. But when Hadoop software is used, companies are saving lots of costs of maintaining the data.
3. So, extra data can be stored at the same price if we use Hadoop instead of using traditional method of handling big data.
4. This allows companies to capture more and more data at a low cost.

E. Provides scalable analytics :

1. The HDFS and Map Reduce components of Hadoop allow parallel storing and processing of data.
2. With the increase of volume of the data the analytics can be scalable in parallel distributed way.

F. Provides rich analytics :

1. Hadoop has a unique quality of handling big data in different programming languages.
2. The project in Hadoop can be done using one of these coding languages Java, Python, SQL, R, and Ruby.
3. They are open source and easy to learn the programming languages.

Que 2.9. Which tools are used to analyze data using Hadoop ?

Answer

Following tools are used for data analyzing using Hadoop :

1. Apache Spark :

i. Apache Spark in an open-source processing engine that is designed for ease of analytics operations.

ii. It is a cluster computing platform that is designed to be fast and made for general purpose uses.

iii. Spark is designed to cover various batch applications, Machine Learning, streaming data processing, and interactive queries.

2. MapReduce :

i. MapReduce is just like an algorithm or a data structure that is based on the YARN framework.

ii. The primary feature of MapReduce is to perform the distributed processing in parallel in a Hadoop cluster, which makes Hadoop working so fast because when we are dealing with Big data, serial processing is no more of any use.

3. Apache Hive :

i. Apache Hive is a data warehousing tool that is built on top of the Hadoop.

ii. Hive is one of the best tools used for data analysis on Hadoop.

iii. The query language of used in Hive is known as HQL or HIVEQL.

4. Apache Impala :

i. Apache Impala is an open-source SQL engine designed for Hadoop.

ii. Impala overcomes the speed-related issue in Apache Hive with its faster-processing speed.

iii. Apache Impala uses similar kinds of SQL syntax, ODBC driver, and user interface as that of Apache Hive.

iv. Apache Impala can easily be integrated with Hadoop for data analytics purposes.

5. Apache Mahout :

i. Apache Mahout runs the algorithm on the top of Hadoop, so it is named Mahout.

ii. Mahout is mainly used for implementing various Machine Learning algorithms on Hadoop like classification, Collaborative filtering, Recommendation.

iii. Apache Mahout can implement the Machine algorithms without integration on Hadoop.

PART-4

Scaling Out, Hadoop Streaming, Hadoop Pipes, Hadoop Echo System.

Questions-Answers

Long Answer Type and Medium Answer Type Questions

Que 2.10. Describe the term scaling out.

OR

Differentiate "Scale up and scale out". Explain with an example how Hadoop uses scale out feature to improve the performance.

Answer

S. No.	Scale up	Scale out
1.	The term "scaling-up" means to use a powerful single server to process the workload that fits within the server boundaries.	Scale-out utilizes multiple processors as a single entity so we can scale beyond the computer capacity of a single server.
2.	Scale-up system consists of many shelves of drives and a pair of controllers.	Scale-out systems consists of clusters, which are co-equal nodes that work together.
3.	As you need more space, you add more shelves of drives.	As you need more space, nodes can be added or removed.
4.	You need to purchase new hardware every time you want to upgrade your system.	You do not have to purchase new hardware every time you want to upgrade your system.

Scale out :

1. To scale out the data flow for large inputs, we need to store the data in a distributed filesystem, typically HDFS.

2. This allows Hadoop to move the MapReduce computation to each machine hosting a part of the data.
3. Hadoop runs the job by dividing it into tasks.
4. There are two types of task: map tasks and reduce tasks.
5. The nodes that control the job execution process are: jobtracker and tasktracker.
6. The jobtracker coordinates all the jobs run on the system by scheduling tasks to run on tasktrackers.
7. Tasktrackers run tasks and send progress reports to the jobtracker.
8. Hadoop divides the input to a MapReduce job into fixed-size pieces called splits.
9. Hadoop creates one map task for each split.
10. The time taken to process each split is small.
11. If we are processing the splits in parallel, the processing is better load-balanced.
12. For most jobs, a good split size tends to be the size of an HDFS block.
13. Hadoop runs the map task on a node where the input data resides in HDFS.

Que 2.11. What is Hadoop Streaming ?

Answer

1. Hadoop Streaming is a utility that comes with the Hadoop distribution.
2. This utility allows us to create and run Map Reduce jobs with any executable or script as the mapper and the reducer.
3. It uses Unix streams as the interface between the Hadoop and our Map Reduce program so that we can use any language which can read standard input and write to standard output to write for writing our Map Reduce program.
4. Hadoop Streaming supports the execution of non-Java, programmed Map Reduce jobs execution over the Hadoop cluster.
5. It supports Python, Perl, R, PHP, and C++ programming languages.

Que 2.12. Write short note on Hadoop Pipes.

Answer

1. Hadoop Pipes is the name of the C++ interface to Hadoop Map Reduce.
2. Unlike streaming, which uses standard input and output to communicate with the map and reduce code, Pipes uses sockets as the channel over which the tasktracker communicates with the process running the C++ map or reduce function.

3. Hadoop pipes allow users to use the C++ language for Map Reduce programming.
4. The main method it takes is to put the C++ code of the application logic in a separate process, and then let the Java code communicate with C++ code through the socket.
5. To a large extent, this approach is similar to Hadoop streaming, where communication differs : one is the standard input output and the other is the socket.

Que 2.13. Describe briefly Hadoop Ecosystem.

Answer

1. Hadoop Ecosystem is a platform or a suite which provides various services to solve the Big data problems.
2. It includes Apache projects and various commercial tools and solutions.
3. Most of the tools or solutions are used to supplement or support these major elements.
4. All these tools work collectively to provide services such as absorption, analysis, storage and maintenance of data etc.
5. Different components of the Hadoop are as follows :

1. HDFS (Hadoop Distributed File System) :

i. It is the storage component of Hadoop that stores data in the form of files.

ii. Each file is divided into blocks of 128MB (configurable) and stores them on different machines in the cluster.

iii. It has a master-slave architecture with two main components : Name Node and Data Node.

2. MapReduce :

i. To handle Big data, Hadoop relies on the MapReduce algorithm.

ii. It essentially divides a single task into multiple tasks and processes them on different machines.

iii. It has two important phases : Map and Reduce.

iv. Map phase filters, groups, and sorts the data.

v. Reduce phase aggregates the data, summarises the result, and stores it on HDFS.

3. YARN :

i. YARN or Yet Another Resource Negotiator manages resources in the cluster and manages the applications over Hadoop.

ii. It allows data stored in HDFS to be processed and run by various data processing engines.

4. HBase :

i. HBase is a Column-based NoSQL database.

ii. It runs on top of HDFS and can handle any type of data.

iii. It allows for real-time processing and random read/write operations to be performed in the data.

5. Pig :

i. Pig was developed for analyzing large datasets and overcomes the difficulty to write map and reduce functions.

ii. It consists of two components : Pig Latin and Pig Engine.

iii. Pig Latin is the Scripting Language.

iv. Pig Engine is the execution engine on which Pig Latin runs.

5. Hive :

i. Hive is a distributed data warehouse system.

ii. It allows for easy reading, writing, and managing files on HDFS.

iii. It has its own querying language known as Hive Querying Language (HQL).

6. Sqoop :

i. Sqoop plays an important part in bringing data from Relational Databases into HDFS.

ii. The commands written in Sqoop internally converts into MapReduce tasks that are executed over HDFS.

iii. It works with almost all relational databases.

iv. It can also be used to export data from HDFS to RDBMS.

7. Flume :

i. Flume is an open-source service used to efficiently collect, aggregate, and move large amounts of data from multiple data sources into HDFS.

ii. It can collect data in real-time as well as in batch mode.

iii. It has a flexible architecture and is fault-tolerant with multiple recovery mechanisms.

8. Kafka :

i. Kafka sits between the applications generating data (Producers) and the applications consuming data (Consumers).

ii. Kafka is distributed and has in-built partitioning, replication, and fault-tolerance.

iii. It can handle streaming data and also allows businesses to analyze data in real-time.

9. **Oozie :**
   i. Oozie is a workflow scheduler system that allows users to link jobs written on various platforms like MapReduce, Hive, Pig, etc.
   ii. Using Oozie you can schedule a job in advance and can create a pipeline of individual jobs to be executed sequentially or in parallel to achieve a bigger task.
10. **Zookeeper :**
   i. In a Hadoop cluster, coordinating and synchronizing nodes can be a challenging task.
   ii. Zookeeper is the perfect tool for the problem.
   iii. It is an open-source, distributed, and centralized service for maintaining configuration information, naming, providing distributed synchronization, and providing group services across the cluster.

PART-5

Map Reduce, Map Reduce Framework and Basics, How Map Reduce Works.

Questions-Answers

Long Answer Type and Medium Answer Type Questions

Que 2.14. Write short note on MapReduce.

Answer

1. MapReduce is a framework using which we can write applications to process huge amounts of data, in parallel, on large clusters of commodity hardware in a reliable manner.
2. MapReduce is a processing technique and a program model for distributed computing based on Java.
3. The MapReduce algorithm contains two important tasks, namely Map and Reduce.
4. Map takes a set of data and converts it into another set of data, where individual elements are broken down into tuples (key/value pairs).
5. Reduce task, which takes the output from a map as an input and combines those data tuples into a smaller set of tuples.

6. As the sequence of the name MapReduce implies, the reduce task is always performed after the map job.

Que 2.15. Explain different phases of MapReduce.

OR

What is MapReduce ? Explain the stages of MapReduce program execution.

Answer

MapReduce : Refer Q. 2.14, Page 2–14Q, Unit-2.

Following are the different phases/stages of MapReduce program execution :

1. **Input splits :**
   i. An input to a MapReduce in Big data job is divided into fixed-size pieces called input splits.
   ii. Input split is a chunk of the input that is consumed by a single map.
2. **Mapping :**
   i. In this phase data in each split is passed to a mapping function to produce output values.
   ii. For example, a job of mapping phase is to count a number of occurrences of each word from input splits and prepare a list in the form of <word, frequency>.
3. **Shuffling :**
   i. This phase consumes the output of Mapping phase.
   ii. Its task is to consolidate the relevant records from Mapping phase output.
   iii. For example, the same words are clubbed together along with their respective frequency.
4. **Reducing :**
   i. In this phase, output values from the shuffling phase are aggregated.
   ii. This phase combines values from shuffling phase and returns a single output value.

Que 2.16. Describe how MapReduce works.

OR

Explain in detail about MapReduce workflows.

Answer

Steps of MapReduce workflows :

1. **Input Files :**
   i. In input files data for MapReduce job is stored.
   ii. Input files reside in HDFS.

iii. Input files format is arbitrary. Line-based log files and binary format can also be used.

2. **InputFormat :**
   i. InputFormat defines how to split and read these input files.
   ii. It selects the files or other objects for input.
   iii. InputFormat creates InputSplit.

3. **InputSplits :**
   i. It represents the data which will be processed by an individual Mapper.
   ii. For each split, one map task is created. Thus the number of map tasks is equal to the number of InputSplits.
   iii. Framework divide split into records, which mapper process.

4. **RecordReader :**
   i. It communicates with the InputSplit.
   ii. And then converts the data into key-value pairs suitable for reading by the Mapper.

5. **Mapper :**
   i. It processes input record produced by the RecordReader and generates intermediate key-value pairs.
   ii. The intermediate output is completely different from the input pair.
   iii. The output of the mapper is the full collection of key-value pairs.
   iv. The Mapper passes the output to the combiner for further processing.

6. **Combiner :**
   i. Combiner performs local aggregation on the mapper's output.
   ii. It minimizes the data transfer between mapper and reducer.
   iii. When the combiner functionality completes, framework passes the output to the partitioner for further processing.

7. **Partitioner :**
   i. Partitioner comes into the existence if we are working with more than one reducer.
   ii. It takes the output of the combiner and performs partitioning.
   iii. Partitioning in MapReduce execution allows even distribution of the map output over the reducer.

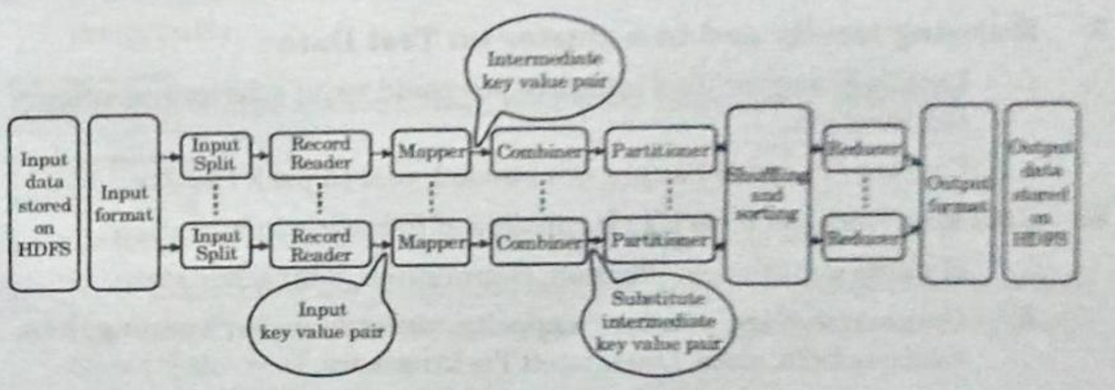
8. **Shuffling and Sorting :**
   i. After partitioning, the output is shuffled to the reduce node.
   ii. The shuffling is the physical movement of the data which is done over the network.

   iii. As all the mappers finish and shuffle the output on the reducer nodes.
   iv. Then framework merges this intermediate output and sort. This is then provided as input to reduce phase.

9. **Reducer :**
   i. Reducer then takes set of intermediate key-value pairs produced by the mappers as the input.
   ii. After that runs a reducer function on each of them to generate the output.
   iii. The output of the reducer is the final output. Then framework stores the output on HDFS.

10. **RecordWriter :**
    i. It writes these output key-value pair from the Reducer phase to the output files.

11. **OutputFormat :**
    i. OutputFormat defines the way how RecordReader writes these output key-value pairs in output files.
    ii. The OutputFormat instances write the final output of reducer on HDFS.



PART-6

Developing a Map Reduce Application, Unit Test with MR Unit, Test Data and Local Tests, Anatomy of a Map Reduce Job Run.

Questions-Answers

Long Answer Type and Medium Answer Type Questions

Que 2.17. **Give the phases of developing a MapReduce application.**

Answer

Phases of developing a MapReduce Application :

1. **Configuration API :** A Configuration class is used to access the configuration XML and can be combined (if a var is repeated, last is used). Variables can also be expanded using system properties.
2. **Configuring the Development Environment :** All JAR's from top level Hadoop directory must be added to the IDE. Also, you can have local and cluster file configurations.
3. **GenericOptionsParser, Tool and ToolRunner :**
   i. GenericOptionsParser interprets Hadoop command-line options and sets them on a Configuration object.
   ii. Tool is an interface to use the above class.
4. **Writing Unit Tests :**
   i. **Mapper Unit Test :** Because Mapper and Reducers writes to Context files (instead of returning the result) a mock for the Context object is needed. We create the context object passing to the static mock method the class. Then we use it normally.
   ii. **Reducer Unit Test :** Reducer unit test is similar to mapper unit test.
5. **Running locally and in a cluster on Test Data :**
   i. Locally Using the Tool interface you could write a driver to configure the local job.
   ii. Cluster No code changes are needed, just to pack the Jar.
6. **The MapReduce Web UI :** It consists of following information :
   i. Hadoop installation : version, compilation, jobtracker state.
   ii. Summary of the cluster : capacity, utilization, mr running, jobs, tasktrackers, slots, blacklisted Tasktrackers.
   iii. Job Scheduler : Running and failed jobs with id's, owner, name.
   iv. Link to Jobtracker Logs : historic.
7. **Hadoop Logs :** Logfiles can be found on the local fs of each TaskTracker and if JVM reuse is enabled, each log accumulates the entire JVM run. Anything written to standard output or error is directed to the relevant logfile.
8. **Tuning a Job to improve performance :** After the program is working, you may wish to do some tuning, first by running through some standard checks for making MapReduce programs faster and then by doing task profiling.

Que 2.18. How to write a program for MapReduce application ?

Answer

1. Writing a program in MapReduce follows a certain pattern.
2. You start by writing your map and reduce functions, ideally with unit tests to make sure they do what you expect.
3. Then you write a driver program to run a job, which can run from your IDE using a small subset of the data to check that it is working.
4. If it fails, you can use your IDE's debugger to find the source of the problem.
5. With this information, you can expand your unit tests to cover this case and improve your mapper or reducer to handle such input correctly.
6. When the program runs as expected against the small dataset, you are ready to unleash it on a cluster.
7. Running against the full dataset is likely to expose some more issues, which you can fix as before, by expanding your tests and mapper or reducer to handle the new cases.
8. After the program is working, you may wish to do some tuning, first by running through some standard checks for making MapReduce programs faster and then by doing task profiling.
9. Profiling distributed programs is not easy, but Hadoop has hooks to aid the process.

Que 2.19. Write a short note on : Unit tests with MRUnit.

Answer

1. Testing and debugging multi threaded programs is hard.
2. Now take the same programs and massively distribute them across multiple JVMs deployed on a cluster of machines and the complexity goes off the roof.
3. One way to overcome this complexity is to do testing in isolation and catch as many bugs as possible locally.
4. MRUnit is a testing framework that lets you test and debug MapReduce jobs in isolation without spinning up a Hadoop cluster.
5. MRUnit provides a powerful and light-weight approach to do test-driven development.
6. This makes it easy to develop as well as to maintain Hadoop MapReduce code bases.
7. MRUnit supports testing Mappers and Reducers separately as well as testing MapReduce computations as a whole.
8. MRUnit allows you to do TDD (Test Driven Development) and write lightweight unit tests which accommodate Hadoop's specific architecture and constructs.

Que 2.20. **Write a short note on : Test data and local tests in MapReduce.**

Answer

1. After getting the mapper and reducer working on controlled inputs, the next step is to write a job driver.
2. This job driver is then executed on some test data on a development environment.
3. To run a job, Hadoop comes with a local job runner.
4. It is a cut-down version of the MapReduce execution engine for running MapReduce jobs in a single JVM.
5. It is designed for testing and is very convenient for use in an IDE.
6. However this local job runner is only designed for simple testing of MapReduce programs, so it differs from full MapReduce implementation.
7. The main difference is that it can't run more than one reducer.
8. The local job runner is enabled by a configuration setting.

Que 2.21. **How does Hadoop executes a MapReduce program ?**

Answer

1. We can run a MapReduce job with a single method call: submit() on a Job object.
2. Now Hadoop executes a MapReduce program depending on following configuration settings :

A. Hadoop up to 0.20 release series :

i. In releases of Hadoop up to 0.20 release series, mapred.job.tracker determines the means of execution.

ii. If this configuration property is set to local then the local job runner is used.

iii. If this configuration property is set to a colon-separated host and port pair, then the property is interpreted as a jobtracker address.

B. Hadoop 0.23.0 release series :

i. In Hadoop 0.23.0, MapReduce 2 implementation was introduced.

ii. It is built on a system called YARN.

iii. In this configuration property takes the values local (for the local job runner), classic (for the "classic" MapReduce framework), and yarn (for the new framework).

Que 2.22. **Explain anatomy of job run in classic MapReduce (MapReduce 1).**

Answer

1. A job run in classic MapReduce is shown in Fig. 2.22.1.
2. On the top level, there are four independent entities : client, jobtracker, tasktrackers, and distributed filesystem.

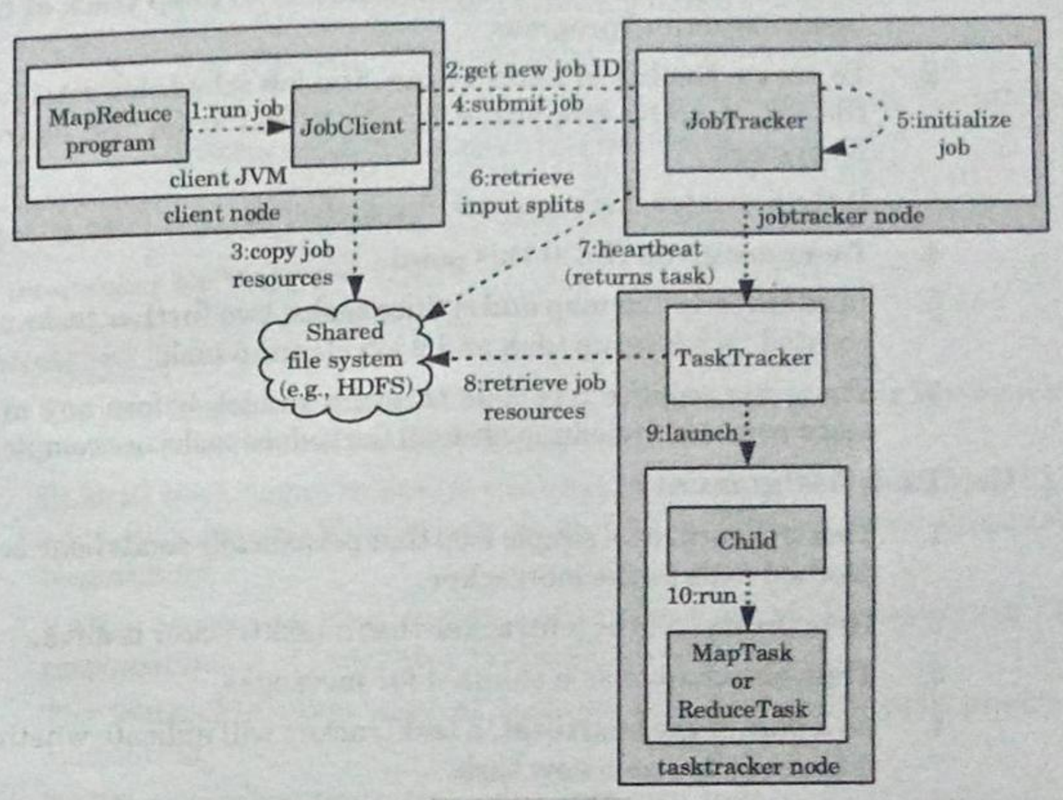


Fig. 2.22.1.

3. The job run in classic MapReduce consists of following :

A. Job Submission :

1. MapReduce program is submitted to JobClient which will then be submitted to HDFS.
2. After submitting the job, the job's progress is polled once every second.
3. This progress is then reported to the console.
4. When the job is completed successfully the job counters are displayed.
5. If the job fails, the error that caused the job to fail is logged to the console.
6. The job submission process does the following :
   i. Asks the jobtracker for a new job ID.

ii. Checks the output specification of the job.

iii. Computes the input splits for the job.

iv. Copies the resources needed to run the job to the jobtracker's filesystem.

v. Tells the jobtracker that the job is ready for execution.

B. Job Initialization :

1. Initialization involves creating an object to represent the job being run and bookkeeping information to keep track of the tasks status and progress.
2. To create the list of tasks to run, the job scheduler retrieves the input splits computed by the client from the shared filesystem.
3. It then creates one map task for each split.
4. Tasks are given IDs at this point.
5. In addition to the map and reduce tasks, two further tasks are created : a job setup task and a job cleanup task.
6. These are used to run code to setup the job before any map tasks run, and to cleanup after all the reduce tasks are complete.

C. Task Assignment :

1. Tasktrackers run a simple loop that periodically sends heartbeat method calls to the jobtracker.
2. Heartbeats tell the jobtracker that a tasktracker is alive.
3. They also double as a channel for messages.
4. As a part of the heartbeat, a tasktracker will indicate whether it is ready to run a new task.
5. If it is ready to run a new task, the jobtracker will allocate it a task.
6. This will be communicated to the tasktracker using the heartbeat return value.

D. Task Execution :

1. Now that the tasktracker has been assigned a task, the next step is for it to run the task.
2. First, it localizes the job JAR by copying it from the shared filesystem to the tasktracker's filesystem.
3. It also copies any files needed from the distributed cache by the application to the local disk.
4. Second, it creates a local working directory for the task, and un-jars the contents of the JAR into this directory.
5. Third, it creates an instance of TaskRunner to run the task.

6. TaskRunner launches a new Java Virtual Machine to run each task.

E. Job Completion :

1. When the jobtracker receives a notification that the last task for a job is complete it changes the status of the job to "successful."
2. When the jobtracker learns that the job has completed successfully it prints a message to tell the user.
3. The jobtracker also sends an HTTP job notification.
4. Last, the jobtracker cleans up its working state for the job and instructs tasktrackers to do the same.

Que 2.23. How the scalability shortcomings of classic MapReduce is overcome by YARN ?

Answer

1. For very large clusters (4000 nodes and higher), the classic MapReduce system faces issue of scalability bottlenecks.
2. In 2010 work began to design the next generation of MapReduce. This next generation MapReduce was YARN (Yet Another Resource Negotiator).
3. YARN overcame the scalability shortcomings by splitting the responsibilities of jobtracker into separate entities.
4. The jobtracker takes care of both job scheduling and task progress monitoring.
5. YARN separates these two roles into two independent daemons : a resource manager and an application master.
6. The resource manager manages the use of resources across the cluster.
7. The application master manages the lifecycle of applications running on the cluster.

Que 2.24. Explain anatomy of job run in YARN (MapReduce 2).

Answer

1. A job run in YARN (MapReduce 2) is shown in Fig. 2.24.1.
2. On the top level, there are five independent entities: client, YARN resource manager, YARN node managers, MapReduce application master, and distributed filesystem.

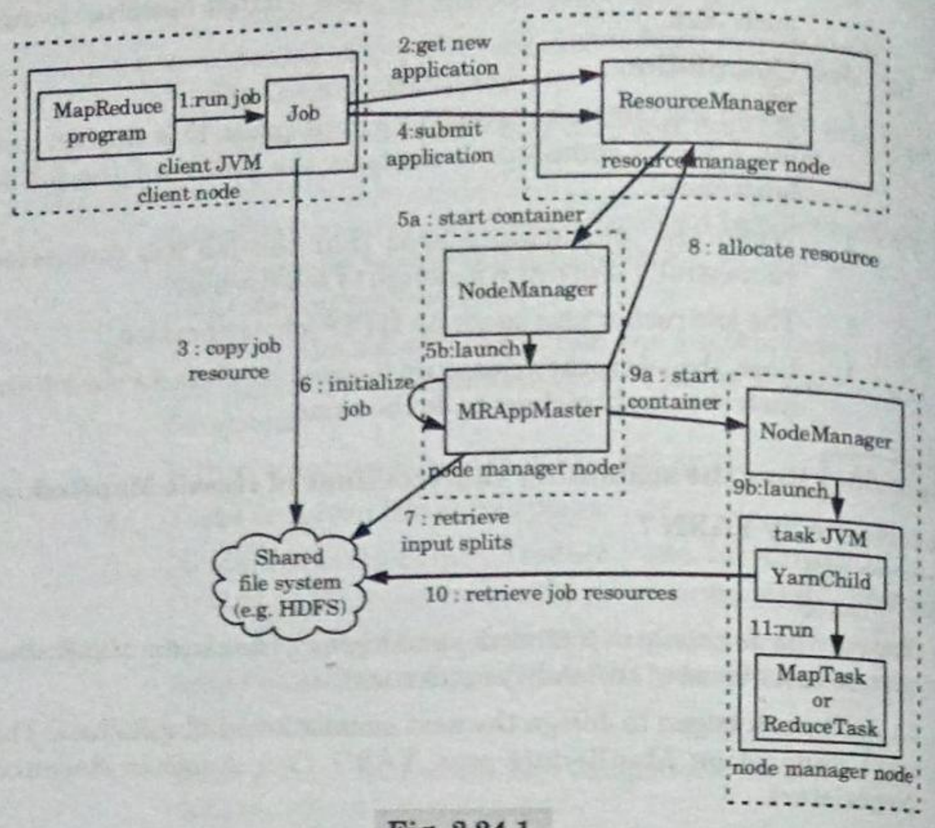


Fig. 2.24.1.

3. The job run in YARN (MapReduce 2) consists of following :

A. Job Submission :

1. Jobs are submitted in MapReduce 2 using the same user API as classic MapReduce.
2. ClientProtocol in MapReduce 2 is activated when mapreduce.framework.name is set to yarn.
3. The new job ID is retrieved from the resource manager.
4. The job client checks the output specification of the job; computes input splits and copies job resources to HDFS.
5. Finally, the job is submitted on the resource manager.

B. Job Initialization :

1. When the resource manager receives a call it hands off the request to the scheduler.
2. The scheduler allocates a container.
3. Under the node manager's management the resource manager then launches the application master's process.

4. The application master initializes the job by creating a number of bookkeeping objects to keep track of the job's progress.
5. Next, it retrieves the input splits computed in the client from the shared filesystem.
6. It then creates a map task object for each split, and a number of reduce task objects.
7. After this the application master decides how to run the tasks that make up the MapReduce job.
8. If the job is small, the application master may choose to run them in the same JVM as itself.
9. Such a job runs as an uber task.

C. Task Assignment :

1. If the job is not running as an uber task, then the application master requests containers for all the map and reduce tasks from the resource manager.
2. Each request includes information about each map task's data locality.
3. The scheduler uses this information to make scheduling decisions.
4. In an ideal case it attempts to place tasks on data-local nodes.
5. Requests also specify memory requirements for tasks.

D. Task Execution :

1. Once a task has been assigned a container, the application master starts the container by contacting the node manager.
2. The task is executed by a Java application.
3. Before the Java application can run the task it localizes the resources that the task needs.
4. Finally, it runs the map or reduce task.

E. Job Completion :

1. Every five seconds the client poll the application master for progress and checks whether the job has completed.
2. This polling interval can be set using configuration property.
3. It also supports notification of job completion via an HTTP callback.
4. On job completion the application master and the task containers clean up their working state.

PART-7

Failures, Job Scheduling, Shuffle and Sort, Task Execution, MapReduce Types, Input Formats, Output Formats, Map Reduce Features, Real-World Map Reduce.

Questions-Answers

Long Answer Type and Medium Answer Type Questions

Que 2.25. What are various failures in classic MapReduce (MapReduce 1) ?

Answer

In classic MapReduce (MapReduce 1) following are the three failures modes :

A. Task Failure :

1. Tasktracker marks tasks as failed in following situations :
   i. User code in the map or reduce task throws a runtime exception.
   ii. If streaming process exits with a nonzero exit code.
   iii. Sudden exit of child JVM.
   iv. Tasktracker notices that it has not received progress update for a while and proceeds to mark the task as failed.
2. When jobtracker is notified of task attempt that has failed, it will reschedule execution of the task.
3. If the task fails four times or more, it will not be retried further.
4. User may also kill or fail task attempts using the Web UI or the command line.

B. Tasktracker Failure :

1. Failure of tasktracker is another failure mode.
2. If tasktracker fails by crashing, or running very slowly, it will stop sending heartbeats to jobtracker.
3. Jobtracker will notice tasktracker that has stopped sending heartbeats and remove it from its pool of tasktrackers to schedule tasks on.

4. Tasktracker can also be blacklisted by jobtracker, even if the tasktracker has not failed.
5. If more than four tasks from the same job fail on particular tasktracker, then the jobtracker records this as a fault.
6. Blacklisted tasktrackers are not assigned tasks, but they continue to communicate with the jobtracker.
7. Faults expire overtime (at rate of one per day), so tasktrackers get chance to run jobs again simply by leaving them running.

C. Jobtracker Failure :

1. It is the most serious failure mode.
2. Hadoop has no mechanism for dealing with failure of the jobtracker.
3. It is a single point of failure, so in this case the job fails.
4. However, this failure mode has a low chance of occurring.

Que 2.26. What are various failures in YARN (MapReduce 2) ?

Answer

In YARN (MapReduce 2) following are the four failures modes :

A. Task Failure :

1. Failure of the running task is similar to the classic case.
2. The tasks are marked as failed in following situations :
   i. Runtime exceptions and sudden exits of the JVM are propagated back to the application master.
   ii. If hanging tasks are noticed by the application master by the absence of a ping over the umbilical channel.
   iii. A task is marked as failed after four attempts.

B. Application Master Failure :

1. An application master sends periodic heartbeats to the resource manager.
2. In the event of application master failure, the resource manager will detect the failure.
3. The resource manager will start a new instance of the application master running in a new container.
4. Also the client polls the application master for progress reports.
5. If its application master fails the client needs to locate the new instance.

6. The client will go back to the resource manager to ask for the new application master's address.

C. Node Manager Failure :

1. If a node manager fails, then it will be removed from the resource manager's pool of available nodes.
2. Node managers may be blacklisted if the number of failures for the application is high.
3. Blacklisting is done by the application master.

D. Resource Manager Failure :

1. If the resource manager fails, then neither jobs nor task containers can be launched.
2. The resource manager was designed to be able to recover from crashes, by using a checkpointing mechanism to save its state to persistent storage.
3. The state consists of the node managers in the system and the running applications.

Que 2.27. What are the types of schedulers in MapReduce ?

Answer

Following are the types of schedulers in MapReduce :

1. **Capacity scheduler :**
   i. In capacity scheduler, we have multiple job queues for scheduling our tasks.
   ii. The capacity scheduler allows multiple occupants to share a large size Hadoop cluster.
   iii. In capacity scheduler corresponding for each job queue, we provide some slots or cluster resources for performing job operation.
   iv. Each job queue has its own slots to perform its task.
   v. In case we have tasks to perform in only one queue then the tasks of that queue can access the slots of other queues also as they are free to use, and when the new task enters to some other queue then jobs in running in its own slots of the cluster are replaced with its own job.
2. **Fair scheduler :**
   i. The Fair scheduler is similar to that of the capacity scheduler. The priority of the job is kept in consideration.

   ii. With the help of Fair scheduler, the YARN applications can share the resources in the large Hadoop Cluster and these resources are maintained dynamically so no need for prior capacity.
   iii. The resources are distributed in such a manner that all applications within a cluster get an equal amount of time.
   iv. Fair scheduler takes scheduling decision on the basis of memory; we can configure it to work with CPU also.

Que 2.28. What are the advantages and disadvantages of different types of scheduler ?

Answer

Following are the advantages and disadvantages of different types of scheduler :

Advantage of Capacity scheduler :

1. Best for working with multiple clients or priority jobs in a Hadoop cluster.
2. Maximizes throughput in the Hadoop cluster.

Disadvantage of Capacity scheduler :

1. More complex.
2. Not easy to configure for everyone.

Advantage of Fair scheduler :

1. Resources assigned to each application depend upon its priority.
2. It can limit the concurrent running task in a particular queue.

Disadvantage of Fair scheduler :

1. The configuration is required.

Que 2.29. What is shuffle and sort in Hadoop MapReduce ?

Answer

1. Shuffle phase in Hadoop transfers the map output from Mapper to a Reducer in MapReduce.
2. Sort phase in MapReduce covers the merging and sorting of map outputs.
3. Data from the mapper are grouped by the key, split among reducers and sorted by the key.
4. Every reducer obtains all values associated with the same key.
5. Shuffle and sort phase in Hadoop occur simultaneously and are done by the MapReduce framework.

A. Shuffle in MapReduce :

1. The process by which the system performs the sort and transfers the map output to the reducer as input is known as shuffle.
2. MapReduce shuffle phase is necessary for the reducers; otherwise, they would not have any input.
3. As shuffle can start even before the map phase has finished so this saves some time and completes the tasks in lesser time.

B. Sort in MapReduce :

1. The keys generated by the mapper are automatically sorted by MapReduce Framework.
2. Sorting in Hadoop helps reducer to easily distinguish when a new reduce task should start.
3. This saves time for the reducer.
4. Reducer starts a new reduce task when the next key in the sorted input data is different than the previous.
5. Each reduce task takes key-value pairs as input and generates key-value pair as output.

Que 2.30. Write a short note on :

i. Speculative Execution.
ii. Task JVM Reuse.
iii. Skipping Bad Records.

Answer

A. Speculative Execution :

1. The MapReduce model break jobs into tasks and run these tasks in parallel to make the overall job execution time smaller.
2. This makes job execution time sensitive to slow-running tasks.
3. Tasks may be slow for various reasons, including hardware degradation or software misconfiguration.
4. Hadoop doesn't try to diagnose and fix slow-running tasks.
5. Instead, it tries to detect when a task is running slower and launches another, equivalent, task as a backup.
6. This is termed as speculative execution of tasks.
7. A speculative task is launched only for tasks that have been running for some time (at least a minute) and have failed to make as much progress as the other tasks from the job.
8. Speculative execution is an optimization.
9. It is not a feature to make jobs run more reliably.

B. Task JVM Reuse :

1. Starting a new JVM for each task can take around a second.
2. This is insignificant for jobs that run for a minute or so.
3. However, jobs that have a large number of very short-lived tasks can achieve significant performance gains if the JVM is reused for subsequent tasks.
4. When the JVM is reused the JVM runs tasks sequentially.
5. Tasks that are CPU-bound may also benefit from task JVM reuse.
6. These tasks take advantage of runtime optimizations applied by the HotSpot JVM.
7. Another place where a shared JVM is useful is for sharing state between the tasks of a job.

C. Skipping Bad Records :

1. Large datasets often have corrupt records.
2. If a small percentage of records are corrupt, then skipping them may not significantly affect the result.
3. We can use Hadoop's optional skipping mode for automatically skipping bad records.
4. When skipping mode is enabled, tasks report the records being processed back to the tasktracker.
5. When the task fails, the tasktracker retries the task, skipping the records that caused the failure.
6. Skipping mode is turned on for a task only after it has failed twice.
7. Skipping mode can detect only one bad record per task attempt, so this mechanism is appropriate only for detecting occasional bad records.

Que 2.31. What are the different types of input formats in Hadoop ?

Answer

Following the different input formats available :

A. Input Splits and Records :

1. An input split is a chunk of the input that is processed by a single map.
2. Each map processes a single split.
3. Each split is divided into records, and the map processes each record in turn.

4. Splits and records are logical.
5. In a database context, a split might correspond to a range of rows from a table and a record to a row in that range.

B. Text Input : Following are different InputFormats that Hadoop provides to process text :

1. **TextInputFormat :** TextInputFormat is the default InputFormat. Each record is a line of input. The key is the byte offset within the file of the beginning of the line. The value is the contents of the line and is packaged as a Text object.
2. **KeyValueTextInputFormat :** It is common for each line in a file to be a key-value pair, separated by a delimiter. To interpret such files correctly, KeyValueTextInputFormat is appropriate.
3. **NLineInputFormat :** If you want your mappers to receive a fixed number of lines of input, then NLineInputFormat is used.

C. Binary Input : Following are different binary formats Hadoop MapReduce supports :

1. **SequenceFileInputFormat :** To use data from sequence files as the input to MapReduce, you use SequenceFileInputFormat. The keys and values are determined by the sequence file, and you need to make sure that your map input types correspond.
2. **SequenceFileAsTextInputFormat :** It is a variant of SequenceFileInputFormat that converts the sequence file's keys and values to Text objects. This format makes sequence files suitable input for Streaming.
3. **SequenceFileAsBinaryInputFormat :** It is a variant of SequenceFileInputFormat that retrieves the sequence file's keys and values as opaque binary objects.

D. Multiple Inputs :

1. Over time the data format evolves. So we have to write our mapper to cope with all of our legacy formats.
2. Or, we have data sources that provide the same type of data but in different formats.
3. These cases are handled easily by using the MultipleInputs class, which allow us to specify the InputFormat and Mapper to use on a per-path basis.

E. Database Input (and Output) :

1. DBInputFormat is an input format for reading data from a relational database, using JDBC.

Que 2.32. What are the different types of output formats in Hadoop ?

Answer

Following the different output formats available :

A. Text Output :

1. The default output format, TextOutputFormat, writes records as lines of text.
2. Its keys and values may be of any type.
3. Each key-value pair is separated by a tab character.

B. Binary Output : Following are different binary formats Hadoop MapReduce supports :

1. **SequenceFileOutputFormat :** It writes sequence files for its output. This is a good choice of output if it forms the input to a further MapReduce job, since it is compact and is readily compressed.
2. **SequenceFileAsBinaryOutputFormat :** It is the counterpart to SequenceFileAsBinaryInput Format, and it writes keys and values in raw binary format into a SequenceFile container.
3. **MapFileOutputFormat :** It writes MapFiles as output. The keys in a MapFile must be added in order, so you need to ensure that your reducers emit keys in sorted order.

C. Multiple Outputs :

1. There is sometimes a need to have more control over the naming of the files or to produce multiple files per reducer.
2. MapReduce comes with the MultipleOutputs class to help you do this.
3. MultipleOutputs allows you to write data to files whose names are derived from the output keys and values.
4. This allows each reducer to create more than a single file.

D. Lazy Output :

1. Some applications prefer that empty files not be created, which is where Lazy OutputFormat helps.
2. It is a wrapper output format that ensures that the output file is created only when the first record is emitted for a given partition.

E. Database Output :

1. DBOutputFormat is the output format which is useful for dumping job outputs into a database.

Que 2.33. What are the various advanced features of MapReduce?

Answer

Following are the various advanced features of MapReduce :

A. Counters :

1. Counters are a useful channel for gathering statistics about the job.
2. They are also useful for problem diagnosis.
3. Following are various types of counters :

a. Built-in Counters :

1. Hadoop maintains some built-in counters for every job which report various metrics for our job.
2. Several groups for the built-in counters are MapReduce Task Counters, Filesystem Counters, FileInput-Format Counters, FileOutput-Format Counters, Job Counters etc.

b. User-Defined Java Counters :

1. MapReduce allows user code to define a set of counters which are then incremented as desired in the mapper or reducer.
2. Counters are defined by a Java enum, which serves to group related counters.
3. A job may define an arbitrary number of enums, each with an arbitrary number of fields.

c. User-Defined Streaming Counters :

1. A Streaming MapReduce program can increment counters by sending a specially formatted line to the standard error stream.

B. Sorting :

1. The ability to sort data is at the core of MapReduce.
2. Even if application isn't concerned with sorting, it may be able to use the sorting stage to organize its data.
3. Following are different ways of sorting datasets that MapReduce provides :

i. Preparation

ii. Partial Sort

iii. Total Sort

iv. Secondary Sort

C. Joins :

1. MapReduce can perform joins between large datasets.
2. How we implement the join depends on how large the datasets are and how they are partitioned.
3. If the join is performed by the mapper, it is called a map-side join.
4. If it is performed by the reducer, it is called a reduce-side join.
5. If both datasets are too large, then we can still join them using MapReduce, depending on how the data is structured.

☺☺☺