**CS334-Big Data Analytics**

**Unit-1**

**Remembering:**

1. What is the definition of big data?

- Big data refers to large volumes of structured and unstructured data that cannot be easily processed using traditional database management systems.

2. Give an example of unstructured data.

- Examples of unstructured data include text documents, emails, social media posts, or multimedia content.

3. Can you recall an industry that extensively uses big data?

- Industries such as e-commerce, finance, healthcare, telecommunications, and transportation heavily rely on big data.

4. What is the purpose of web analytics?

- Web analytics is used to collect and analyze data related to website usage, visitor behavior, and online marketing campaigns.

**Understanding:**

5. How do big data technologies contribute to analyzing large datasets?

- Big data technologies provide scalable and distributed processing capabilities, allowing efficient analysis of large datasets across computer clusters.

6. What are some common applications of big data in businesses?

- Big data applications include predictive analytics, customer segmentation, fraud detection, recommendation systems, sentiment analysis, and supply chain optimization.

7. Explain the role of Hadoop in big data processing.

- Hadoop is an open-source framework that enables distributed storage and processing of big data across multiple computers, providing fault tolerance and scalability.

8. How do open-source technologies relate to big data?

- Open-source technologies like Hadoop and Spark offer cost-effective and accessible solutions for storing, processing, and analyzing large amounts of data.

**Applying:**

9. How can businesses leverage cloud computing for big data management?

- Cloud computing provides scalable and on-demand resources, making it ideal for storing and processing big data, reducing infrastructure costs and improving flexibility.

10. Describe a scenario where mobile business intelligence can be used.

- Mobile business intelligence can be used when executives or managers need access to real-time data and insights while on the move, enabling quick decision-making.

11. How can crowd sourcing analytics benefit market research?

- Crowdsourcing analytics allows companies to gather data and insights from a large group of people, helping with market research, idea generation, and problem-solving.

12. In what ways can inter-firewall analytics enhance network security?

- Inter-firewall analytics can detect potential threats or anomalies by analyzing network traffic and security logs across multiple firewalls, helping in the prevention of security breaches.

**Analyzing:**

13. What are some challenges associated with processing big data?

- Challenges include data storage and management, data quality and veracity, data privacy and security, data integration from various sources, and the need for advanced analytical skills.

14. Compare and contrast Hadoop and Spark in terms of their functionalities.

- Hadoop and Spark are both big data processing frameworks, but Hadoop focuses on distributed storage and batch processing, while Spark offers real-time data processing and in-memory computing.

15. Analyze the benefits and drawbacks of using open-source technologies for big data processing.

- Benefits include cost-effectiveness and community support, while drawbacks may include limited technical support and potential security vulnerabilities.

16. How does real-time analytics differ from traditional batch processing in big data applications?

- Real-time analytics provides immediate insights and actions based on streaming data, while batch processing analyzes data in large volumes at specific intervals, which may result in delayed insights.

**Evaluating:**

17. Assess the impact of big data on business decision-making processes.

- Big data enables data-driven decision making by providing valuable insights into customer behavior, market trends, and operational optimization, leading to improved business outcomes.

18. Evaluate the ethical considerations associated with big data analytics.

- Ethical considerations include issues of data privacy, consent, transparency, bias in algorithms, and responsible use of data, which need to be carefully addressed to ensure fair and responsible data practices.

19. Compare the advantages and disadvantages of cloud-based big data solutions.

- Advantages of cloud-based solutions include scalability and cost-efficiency, while disadvantages may include potential data security risks and dependence on internet connectivity.

**Creating:**

20. Design a mobile business intelligence application that provides real-time sales data and performance metrics for sales representatives.

- The application should have an intuitive user interface, allow for interactive data visualization, and provide alerts and notifications for important updates or changes in sales data.

**Unit-2**

**Remembering:**

1. What is NoSQL?

- NoSQL (Not Only SQL) refers to a class of non-relational database management systems that are designed to handle large volumes of unstructured and semi-structured data.

2. What are the two common data models in NoSQL databases?

- The two common data models in NoSQL databases are the key-value model and the document model.

3. What is a graph database?

- A graph database is a NoSQL database that uses graph structures with nodes, edges, and properties to represent and store data, making it suitable for data with complex relationships.

4. What is a schemaless database?

- A schemaless database is a type of database that does not require a predefined schema or structure for data storage. It allows flexibility in adding or modifying data without strict schema constraints.

**Understanding:**

5. How does a key-value data model work in NoSQL databases?

- In a key-value data model, data is stored as key-value pairs, where each value is associated with a unique key. It allows for fast and efficient retrieval of data using the key.

6. Explain the document data model in NoSQL databases.

- In the document data model, data is stored as self-contained documents, typically in JSON or BSON format. Documents can contain nested structures and can be queried and indexed.

7. How do graph databases represent relationships between data entities?

- Graph databases represent relationships between data entities through nodes (representing entities) and edges (representing relationships between entities), allowing for efficient traversal of complex relationships.

8. What are materialized views in NoSQL databases?

- Materialized views are precomputed views that store the results of complex queries or aggregations in a separate table. They help improve query performance by avoiding the need to compute results on the fly.

**Applying:**

9. How does master-slave replication work in distributed databases?

- In master-slave replication, one node (master) accepts write operations and replicates the changes to one or more nodes (slaves). Slaves can handle read operations, providing scalability and fault tolerance.

10. Give an example of a NoSQL database that uses the master-slave replication model.

- Cassandra is an example of a NoSQL database that uses the master-slave replication model for data distribution and high availability.

11. How does consistency play a role in distributed databases?

- Consistency in distributed databases refers to ensuring that all nodes in the database have the same view of data after a write operation. It ensures data integrity and accuracy across the system.

12. Describe the data model used in Cassandra.

- Cassandra uses a wide-column data model. It organizes data into tables with rows identified by a unique primary key. Each row contains multiple columns, and columns are grouped into column families.

**Analyzing:**

13. Compare and contrast the key-value and document data models.

- The key-value model stores data as simple key-value pairs, while the document model allows for more complex data structures and querying within the document. Both models provide high scalability and flexibility.

14. Analyze the benefits of using graph databases for representing complex relationships.

- Graph databases excel in handling complex relationships as they provide efficient traversal and querying capabilities, making them ideal for use cases like social networks, recommendation systems, and fraud detection.

15. Assess the advantages and disadvantages of schemaless databases.

- Advantages of schemaless databases include flexibility in data modeling and easier handling of evolving schemas. Disadvantages may include potential data integrity issues and increased complexity in querying.

16. Evaluate the use of materialized views for improving query performance in NoSQL databases.

- Materialized views can significantly improve query performance by precomputing results. However, they introduce additional maintenance overhead and may lead to increased storage requirements.

**Evaluating:**

17. Evaluate the suitability of Cassandra for handling high write-intensive workloads.

- Cassandra's distributed architecture and master-slave replication make it well-suited for handling high write-intensive workloads, providing high availability and fault tolerance.

18. Assess the impact of data model choices on the performance and scalability of NoSQL databases.

- The choice of data model can significantly impact the performance and scalability of NoSQL databases. Key-value models provide high performance, while document and graph models offer more flexibility in querying and relationships.

19. Evaluate the advantages and disadvantages of using Cassandra for distributed data storage.

- Advantages of using Cassandra include linear scalability, fault tolerance, and tunable consistency. Disadvantages may include increased complexity in data modeling and lack of support for complex transactions.

**Creating:**

20. Design a data model for a social media platform using a graph database.

- The data model should include nodes representing users, posts, and comments, with edges representing relationships such as friendships, post ownership, and comment replies. The design should allow for efficient querying and traversal of relationships.

**Unit-3**

**Remembering:**

1. What is a MapReduce workflow?

- A MapReduce workflow refers to the sequence of map and reduce tasks performed on a distributed computing framework to process large-scale datasets.

2. What is MRUnit?

- MRUnit is a unit testing framework for MapReduce programs. It provides utilities to write and execute unit tests for MapReduce jobs.

3. What is the purpose of test data in local tests for MapReduce?

- Test data in local tests for MapReduce allows developers to validate the correctness and performance of their MapReduce jobs in a controlled environment before running them on a distributed cluster.

4. What are the components of a MapReduce job run?

- A MapReduce job run consists of three main components: input data, MapReduce program code, and output data.

**Understanding:**

5. Explain the concept of classic MapReduce.

- Classic MapReduce is a programming model for processing large datasets in a parallel and distributed manner. It involves map and reduce functions that operate on key-value pairs to perform data transformations and aggregations.

6. What is YARN in the context of MapReduce?

- YARN (Yet Another Resource Negotiator) is the resource management framework in Hadoop. It manages resources, scheduling, and execution of MapReduce jobs in a distributed environment.

7. What are some common failures that can occur in classic MapReduce?

- Failures in classic MapReduce can include node failures, disk failures, and network failures, which can impact the overall job execution and may require recovery mechanisms.

8. How does job scheduling work in MapReduce?

- Job scheduling in MapReduce involves allocating resources and determining the execution order of map and reduce tasks to optimize resource utilization and job completion time.

**Applying:**

9. Describe the process of shuffle and sort in MapReduce.

- Shuffle and sort in MapReduce involves transferring data between map and reduce tasks based on intermediate keys, sorting the data by keys, and grouping it for the reduce tasks to process.

10. How are tasks executed in a MapReduce job?

- Tasks in a MapReduce job are executed on worker nodes in a distributed cluster. Map tasks operate on input splits, while reduce tasks process the output of map tasks.

11. What are the different types of MapReduce jobs?

- MapReduce jobs can be classified as batch processing jobs, iterative jobs, streaming jobs, or interactive jobs, depending on their processing requirements and data characteristics.

12. Explain the concept of input formats in MapReduce.

- Input formats in MapReduce define how input data is read and parsed by map tasks. They provide a way to handle different types of input data, such as text files, CSV files, or HBase tables.

**Analyzing:**

13. Compare and contrast the classic MapReduce model and the YARN framework.

- Classic MapReduce focuses on data processing, while YARN provides resource management and job execution capabilities. YARN allows for running different processing frameworks, while MapReduce is specific to Hadoop.

14. Analyze the impact of failures on the overall performance of a MapReduce job.

- Failures in MapReduce can lead to job slowdown, data loss, or job failure. Proper recovery mechanisms, fault tolerance, and data replication strategies are essential to mitigate the impact of failures.

15. Assess the benefits of using unit tests with MRUnit for MapReduce development.

- MRUnit allows developers to write unit tests for MapReduce jobs, enabling them to validate the correctness of their code, identify bugs early, and improve overall code quality.

**Evaluating:**

16. Evaluate the importance of data formats in MapReduce input and output.

- Data formats in MapReduce input and output impact the efficiency of data processing, compatibility with other tools, and ease of data integration. Choosing appropriate formats is crucial for efficient data handling.

17. Assess the significance of the shuffle and sort phase in the MapReduce framework.

- The shuffle and sort phase in MapReduce plays a critical role in data transfer between map and reduce tasks, data grouping, and ensuring that the reduce tasks receive data in a sorted manner for processing.

18. Evaluate the impact of job scheduling strategies on the performance of MapReduce jobs.

- Job scheduling strategies affect resource allocation, data locality, and overall job completion time. Efficient job scheduling can improve resource utilization and minimize job execution time.

**Creating:**

19. Design a MapReduce job that processes log files and calculates the total number of occurrences of each unique log entry.

- The design should include mapper and reducer functions that extract log entries, perform counting, and generate output with log entries as keys and their counts as values.

20. Create a custom input format for MapReduce that can read data from a custom data source, such as a NoSQL database or a streaming API.

- The custom input format should implement the necessary logic to read and parse data from the specific data source and provide it to the map tasks in a suitable format for processing.

**Unit-4**

**Remembering:**

1. What is Hadoop Streaming?

- Hadoop Streaming is a utility that allows users to create and run MapReduce jobs with non-Java languages by using standard input and output streams.

2. What is Hadoop Pipes?

- Hadoop Pipes is a C++ library that enables users to create and run MapReduce jobs using C++ as the programming language.

3. What is the design of Hadoop Distributed File System (HDFS)?

- HDFS is designed to store and manage large volumes of data across multiple nodes in a Hadoop cluster, providing fault tolerance, high throughput, and scalability.

4. What is Avro?

- Avro is a data serialization system that provides a compact and efficient way to exchange data between different programming languages and platforms.

**Understanding:**

5. How does Hadoop scale out to handle large-scale data processing?

- Hadoop scales out by distributing data and processing across a cluster of machines, allowing for parallel execution of tasks and efficient utilization of resources.

6. What are the key concepts of Hadoop Distributed File System (HDFS)?

- Key concepts in HDFS include the NameNode (manages the file system namespace), DataNodes (stores the actual data blocks), and the block-based storage model.

7. Explain the Java interface for HDFS.

- The Java interface for HDFS provides classes and methods to interact with HDFS, allowing applications to read, write, and manipulate files stored in HDFS.

8. How does data flow in a Hadoop MapReduce job?

- Data flows through a Hadoop MapReduce job in a sequence of map and reduce phases, where map tasks process input data and produce intermediate results, which are then sorted, shuffled, and processed by reduce tasks.

**Applying:**

9. How does Hadoop handle I/O operations?

- Hadoop provides various classes and APIs for efficient I/O operations, such as reading and writing data to and from HDFS, including input/output formats and serializers.

10. How does Hadoop ensure data integrity in a distributed environment?

- Hadoop ensures data integrity through data replication, where each data block is replicated across multiple DataNodes, and checksums are used to verify the integrity of data during storage and transmission.

11. Describe how compression is used in Hadoop.

- Hadoop supports data compression to reduce storage space and improve data transfer performance. Compression algorithms like GZIP or Snappy can be used for compressing data in Hadoop.

12. How does serialization work in Hadoop?

- Serialization in Hadoop is the process of converting complex data structures into a byte stream to facilitate storage, transmission, and processing. It allows for efficient data interchange between different components of the Hadoop ecosystem.

**Analyzing:**

13. Compare and contrast Avro and file-based data structures in Hadoop.

- Avro provides a compact, schema-based serialization format with rich data types and language independence. File-based data structures, such as SequenceFile or Parquet, offer columnar storage and efficient data retrieval.

14. Analyze the integration between Hadoop and Cassandra.

- The integration between Hadoop and Cassandra allows users to leverage the scalability and fault tolerance of Hadoop for analyzing large datasets stored in Cassandra, combining batch and real-time processing capabilities.

15. Assess the impact of data format selection on Hadoop performance and interoperability.

- The choice of data format in Hadoop can impact storage efficiency, processing speed, and interoperability with other tools and frameworks. Proper selection based on specific use cases is crucial for optimal performance.

**Evaluating:**

16. Evaluate the advantages and disadvantages of using Hadoop Streaming for non-Java MapReduce jobs.

- Hadoop Streaming allows users to use non-Java languages, enabling flexibility and leveraging existing code. However, it may have performance implications and limited access to advanced Hadoop features.

17. Assess the design of Hadoop Distributed File System (HDFS) for handling large-scale data storage and processing.

- HDFS provides fault tolerance, scalability, and high throughput for storing and processing large volumes of data. It optimizes for streaming data access but may have limitations for random access patterns.

18. Evaluate the benefits and challenges of integrating Hadoop with Cassandra for data analysis.

- The integration of Hadoop with Cassandra allows for combining the scalability of Cassandra with the data processing capabilities of Hadoop. However, it introduces complexities in data synchronization, consistency, and managing the integration components.

**Creating:**

19. Design a data processing pipeline using Hadoop and Avro, where data is ingested from multiple sources, transformed, and stored in HDFS.

- The design should include appropriate Avro schemas, data serialization and deserialization logic, and MapReduce jobs for processing and storing the transformed data in HDFS.

20. Create a workflow for integrating Hadoop and Cassandra, where data is extracted from Cassandra, processed in Hadoop, and the results are written back to Cassandra.

- The workflow should include appropriate data extraction mechanisms, Hadoop processing steps, data transformation, and mechanisms to synchronize the processed data back into Cassandra.

**Unit-5**

**Remembering:**

1. What is HBase?

- HBase is a distributed, column-oriented NoSQL database built on top of Apache Hadoop, designed to handle large-scale, structured data.

2. What are some implementations of HBase clients?

- HBase clients can be implemented using programming languages like Java, Python, or REST APIs.

3. Can you provide an example of using HBase?

- An example use case for HBase is storing and querying time-series data, such as sensor readings or log data.

4. What is Pig Latin?

- Pig Latin is a high-level scripting language used in Apache Pig for expressing data transformations and analysis tasks.

**Understanding:**

5. Explain the data model in HBase.

- HBase data model is a sparse, distributed, and persistent multidimensional sorted map that consists of rows, columns, and cell versions.

6. What is Grunt in Pig?

- Grunt is an interactive shell provided by Apache Pig that allows users to interactively run Pig Latin scripts and perform ad-hoc data exploration.

7. Describe the data model in Pig.

- The data model in Pig is based on relations, where data is organized into rows and columns, similar to a table in a relational database.

8. How do you develop and test Pig Latin scripts?

- Pig Latin scripts can be developed and tested using the Grunt shell, Pig scripts in batch mode, or integrated development environments (IDEs) like Apache Zeppelin or Apache Piggybank.

**Applying:**

9. How would you write a Pig Latin script to load data from a CSV file and perform data filtering based on certain conditions?

- LOAD 'input.csv' USING PigStorage(',') AS (col1:chararray, col2:int);

- filtered\_data = FILTER input\_data BY (col2 > 100);

- STORE filtered\_data INTO 'output' USING PigStorage(',');

10. Provide an example of using HBase as a key-value store.

- To put data into HBase: put 'table', 'row1', 'column\_family:column', 'value'

- To get data from HBase: get 'table', 'row1'

11. How can Pig Latin scripts be used to perform data transformations, such as joining datasets or aggregating data?

- Pig Latin provides operators like JOIN and GROUP BY to perform data transformations such as joining datasets based on a common key or aggregating data based on specific criteria.

12. Explain how to use HiveQL to define tables in Hive.

- HiveQL provides syntax to define tables in Hive using the CREATE TABLE statement, specifying column names, data types, and optional constraints.

**Analyzing:**

13. Compare and contrast HBase and traditional relational databases.

- HBase is a NoSQL column-oriented database designed for scalability and high write throughput, while traditional relational databases follow a structured, table-based data model and are typically optimized for complex query processing.

14. Analyze the advantages of using Pig Latin for data processing tasks compared to writing MapReduce programs directly.

- Pig Latin provides a higher level of abstraction and concise syntax, making it easier to express complex data transformations. It also offers optimizations and automatic execution planning, reducing development time and effort.

15. Evaluate the benefits and drawbacks of using HiveQL for querying and analyzing data compared to traditional SQL.

- HiveQL provides a familiar SQL-like syntax for querying data in Hive, making it easier for users with SQL knowledge. However, it may have limitations compared to full-fledged SQL databases in terms of query performance and advanced SQL features.

**Evaluating:**

16. Assess the suitability of HBase for real-time data processing scenarios.

- HBase is suitable for real-time data processing scenarios due to its ability to handle high write throughput, support for rapid data retrieval, and automatic sharding and replication capabilities.

17. Evaluate the advantages and disadvantages of using Pig Latin for data transformations in comparison to other ETL tools or programming languages.

- Pig Latin offers simplicity and ease of use for data transformations, but it may have limitations in terms of performance, lack of advanced analytics features, and learning curve for users not familiar with the language.

18. Assess the benefits and drawbacks of using Hive for querying and analyzing large datasets stored in Hadoop compared to traditional databases.

- Hive provides a scalable and cost-effective solution for analyzing large datasets in Hadoop, but it may have performance limitations, lack real-time query processing, and may require additional data modeling considerations.

**Creating:**

19. Design a Pig Latin script that loads data from multiple input files, joins them based on a common key, and performs aggregation on the joined dataset.

- Load data from files A and B

- Join data using a common key

- Perform aggregation using the GROUP BY operator

- Store the output in a file or HDFS location

20. Create a HiveQL query that calculates the average value of a specific column in a Hive table and groups the results by a different column.

- SELECT column1, AVG(column2) FROM table\_name GROUP BY column1