**SRM Institute of Science and Technology**

**College of Engineering and Technology**

**School of Computing**

SRM Nagar, Kattankulathur – 603203, Chengalpattu District, Tamilnadu **Set A**

**Academic Year: 2023-24 (Even)**

**Test: CLA-T2** **Date: 03-04-2024**

**Course Code & Title:** **21CSE222T BIG DATA TOOLS &TECHNIQUES** **Duration:** 1:30 Hours

**Year & Sem: II Year / IV Sem** **Max. Marks:** 50

**Course Articulation Matrix: *(to be placed)***

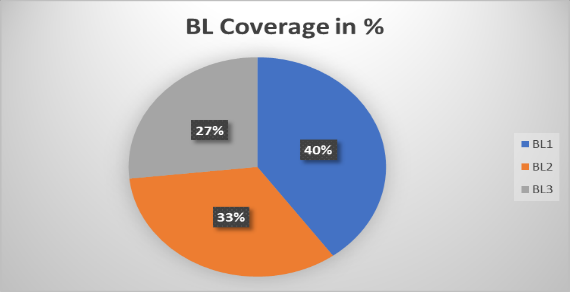
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **S.No.** | **Course Outcome** | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PO12** |
| 1 | **CO1** | - | - | - | - | 1 | - |  | - | - | - | - | - |
| 2 | **CO2** | - | - | - | - | 1 | - | - | - | - | - | - | - |
| 3 | **CO3** | - | - | - | - | 1 | - | - | - | - | - | - | - |
| 4 | **CO4** | - | - | - | - | 1 | - | - | - | - | - | - | - |
| 5 | **CO5** | - | - | - | - | 1 | - | - | - | - | - | - | - |

**Part- A (10 x 1 = 10 Marks)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Q. No** | **Question** | **Marks** | **BL** | **CO** | **PO** |
| 1 | \_\_\_\_\_\_\_\_\_\_ are the work horses of the filesystem.  a) NameNode  b) **DataNode**  c) Job Tracker  d) Task Tracker | **1** | **L1** | **2** | **5** |
| 2 | In Hadoop, inter process communication between nodes in the system is implemented using \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  a)Remote Processing Calls  **b)Remote Procedure Calls**  c)Node Processing Calls  d)Node Procedure Calls | **1** | **L2** | **2** | **5** |
| 3 | A common Hadoop Cluster consists of a \_\_\_\_\_\_\_\_\_\_\_\_ level network topology  a) Two  **b) Three**  c) Four  d) Five | **1** | **L1** | **3** | **5** |
| 4 | Which of the following company has developed PIG?  a) Google  **b) Yahoo**  c) Microsoft  d) Apple | **1** | **L2** | **3** | **5** |
| 5 | \_\_\_\_\_\_\_\_\_\_\_\_ operator in PIG Latin prints a relation to the console  a) LOAD  b) STORE  c) PRINT  **d) DUMP** | **1** | **L1** | **3** | **5** |
| 6 | \_\_\_\_\_\_\_\_\_\_\_ is a data warehouse infrastructure tool to process structured data in Hadoop.  a) PIG  b) SCOOP  **c)** **HIVE**  d) FLUME | **1** | **L1** | **4** | **5** |
| 7 | A Flume agent is a JVM process which has \_\_\_\_\_\_  a) **3 components**  **b) 4 components**  c) 5 components  d) 6 components | **1** | **L2** | **4** | **5** |
| 8 | Oozie workflow runs in which container\_\_\_\_\_\_\_\_\_\_\_  a) Web Servlet  **b) Java Servlet**  c) Apache Servlet  d) XML Servlet | **1** | **L1** | **4** | **5** |
| 9 | The role of the ……………..is to allocate resources across applications.  **a) Cluster manager**  b) Worker Node  c) Executor  d) Task | **1** | **L2** | **4** | **5** |
| 10 | Data Abstraction introduced by Spark SQL is called  a) RDD  b) Schema  c) Relational  **d) SchemaRDD** | **1** | **L2** | **4** | **5** |
|  | **Part B**  **Answer any two (2x5=10 Marks)** |  |  |  |  |
| 11 | What is data integrity in the context of Hadoop I/O?  🡪In the context of Hadoop, data integrity refers to the assurance that data remains accurate and consistent throughout its storage, processing, and retrieval operations. Given the distributed nature of Hadoop and the vast volumes of data it handles, maintaining data integrity is crucial to ensure that no data is lost or corrupted during persistence and processing.  Hadoop employs stringent data integrity constraints to address the challenges posed by disk operations, network transfers, and the sheer volume of data being processed. These constraints include mechanisms such as checksums, replication, fault tolerance, and error detection and correction techniques to safeguard data against errors, corruption, and loss.  Checksums are used to verify the integrity of data blocks during storage and transmission. Replication ensures that multiple copies of data are stored across different nodes in the cluster, reducing the risk of data loss due to hardware failures or disk corruption. Fault tolerance mechanisms enable Hadoop to recover from node failures or data corruption events gracefully, ensuring uninterrupted data availability and reliability. | **5** | **L2** | **2** | **5** |
| 12 | Differentiate between SCOOP & FLUME  🡪Apache Sqoop and Apache Flume work with various kinds of data sources. Flume functions well in streaming data sources which are generated continuously in Hadoop environment such as log files from multiple servers whereas Apache Sqoop is designed to work well with any kind of relational database system that has JDBC connectivity. Sqoop can also import data from NoSQL databases like MongoDB or Cassandra and also allows direct data transfer or Hive or HDFS. For transferring data to Hive using Apache Sqoop tool, a table has to be created for which the schema is taken from the database itself.  In Apache Flume data loading is event driven whereas in Apache Sqoop data load is not driven by events.  Flume is a better choice when moving bulk streaming data from various sources like JMS or Spooling directory whereas Sqoop  is an ideal fit if the data is sitting in databases like Teradata, Oracle, MySQL Server, Postgres or any other JDBC compatible database then it is best to use Apache Sqoop.  In Apache Flume, data flows to HDFS through multiple channels whereas in Apache Sqoop HDFS is the destination for importing data.  Apache Flume has agent based architecture i.e. the code written in flume is known as agent which is responsible for fetching data whereas in Apache Sqoop the architecture is based on connectors. The connectors in Sqoop know how to connect with the various data sources and fetch data accordingly.  Lastly, Sqoop and Flume cannot be used achieve the same tasks as they are developed specifically to serve different purposes. Apache Flume agents are designed to fetch streaming data like tweets from Twitter or log file from the web server whereas Sqoop connectors are designed to work only with structured data sources and fetch data from them.  Apache Sqoop is mainly used for parallel data transfers, for data imports as it copies data quickly where Apache Flume is used for collecting and aggregating data because of its distributed, reliable nature and highly available backup routes. | **5** | **L2** | **3** | **5** |
| 13 | Write the steps for Running an Oozie workflow job A diagram of a map  Description automatically generated  1. Export OOZIE\_URL environment variable which tells the oozie command which Oozie server to use (here we’re using one running locally):  % export OOZIE\_URL="http://localhost:11000/oozie"  2. Run workflow job using-  % oozie job -config ch05/src/main/resources/max-temp-workflow.properties-run  The -config option refers to a local Java properties file containing definitions for the parameters in the workflow XML file, as well as oozie.wf.application.path, which tells Oozie the location of the workflow application in HDFS.  3. Get the status of workflow job-  Status of workflow job can be seen using subcommand ‘job’  with ‘-info’ option and specifying job id after ‘-info’.  e.g., % oozie job -info <job id>  Output shows status which is one of RUNNING, KILLED or  SUCCEEDED. | **5** | **L3** | **4** | **5** |
|  | **Part C**  **Answer any two (15x2=30 Marks)** |  |  |  |  |
| 14 | Your company operates a retail business with stores across multiple regions. Each store maintains transaction data in a structured format, including information about products, sales, and customers. Design a Pig Latin script and outline the steps involved in processing the transaction data and generate summary reports for analysis. Include details about data loading, transformation, aggregation, and storage of the results.  🡪To process transaction data and generate summary reports using Pig Latin, follow these steps:  1. Data Loading: Load transaction data from multiple stores across regions into Pig for further processing. Use the `LOAD` statement to load data from the Hadoop Distributed File System (HDFS) or other supported storage systems. Specify the path to the data file and the appropriate file format loader (e.g., PigStorage for CSV files). Optionally, define the schema of the data using the `AS` clause to ensure proper interpretation of fields.  *example*  *transactions = LOAD 'hdfs://path/to/transaction\_data' USING PigStorage(',') AS (*  *store\_id:int,*  *transaction\_id:int,*  *product\_id:int,*  *quantity:int,*  *amount:double,*  *transaction\_date:chararray );*  2. Data Transformation: Clean and transform the raw transaction data to prepare it for analysis and aggregation. Use Pig Latin operations such as `FILTER`, `FOREACH`, `GENERATE`, `JOIN`, etc., to perform data cleaning and transformation. Handle missing or erroneous data, remove duplicates, standardize formats, and derive new fields if necessary.  *Example:*  *valid\_transactions = FILTER transactions BY amount >= 0;*  3. Aggregation: Aggregate transaction data to generate summary metrics and reports. Use the `GROUP` statement to group data by relevant attributes (e.g., store ID, product category, date). Apply aggregation functions like `SUM`, `AVG`, `MAX`, etc., to compute summary metrics based on grouped data. *example:*  *sales\_by\_store = FOREACH (GROUP valid\_transactions BY store\_id) GENERATE*  *group AS store\_id,*  *SUM(valid\_transactions.amount) AS total\_sales;*  4. Report Generation: Store aggregated data as summary reports for analysis and decision-making. Use the `STORE` statement to save the aggregated data to the desired location and file format.  *example: STORE sales\_by\_store INTO 'hdfs://path/to/sales\_summary' USING PigStorage(',');*  5. Execution: Submit the Pig script to the Pig execution engine for processing. Run the Pig script using the Pig command-line interface (CLI), web interface, or programmatically via API. Monitor job execution and review logs for any errors or warnings. | **15** | **L3** | **3** | **5** |
| 15 | You are creating a distributed system that requires coordination across several nodes. How would you build distributed coordination between these nodes using Apache ZooKeeper? Give instances of the different coordinating jobs that ZooKeeper is capable of handling.  🡪 ZooKeeper is a centralized service designed for managing configuration information, naming, providing distributed synchronization, and offering group services, all of which are essential for distributed applications. Implementing these functionalities in distributed systems often requires significant effort to resolve bugs and race conditions that can arise.  ZooKeeper serves as a distributed coordination service used to manage a large set of hosts. Coordinating and managing services in a distributed environment is inherently complex, but ZooKeeper simplifies this with its straightforward architecture and API. This allows developers to focus on their core application logic without being burdened by the complexities of distribution.  Originally developed at Yahoo! to facilitate easy and robust application access, ZooKeeper later became a standard tool used by frameworks like Hadoop and HBase. For instance, Apache HBase uses ZooKeeper to monitor the status of distributed data.  One of ZooKeeper's key functionalities is cluster synchronization, enabling clusters (groups of nodes) to coordinate and maintain shared data using robust synchronization techniques. ZooKeeper itself is a distributed application that provides services for building and managing other distributed applications.  Common services provided by ZooKeeper include:  Naming service: Identifying nodes in a cluster by name, akin to DNS but for nodes.  Configuration management: Providing the latest configuration information for joining nodes.  Cluster management: Facilitating node joining/leaving operations and real-time node status updates within a cluster. | **15** | **L3** | **3** | **5** |
| 16 | Identify and describe five common limitations of Hadoop that your organization is facing. Then, propose solutions or strategies to overcome each of these limitations effectively, ensuring that your Hadoop-based data infrastructure can meet the organization's requirements efficiently.  🡪  **Issue with Small Files**  The introduction of HAR files (Hadoop Archives) aimed to alleviate the problem of having numerous small files that put pressure on the namenode's memory. Sequence files, where filenames serve as keys and file contents as values, effectively address the "small file problem."  Storing files in HBase is a common design pattern to overcome the small file issue with HDFS, as HBase handles smaller files more efficiently.    **Slow Processing Speed**  To address Hadoop's limitation of slow processing speed, Spark introduced in-memory processing of data. This approach eliminates the time spent on data movement between disk and memory, resulting in faster processing.  **Not Easy to Use**  Spark offers an interactive mode that allows developers and users to receive immediate feedback on queries and activities, making it user-friendly. Additionally, Spark simplifies programming with its abundance of high-level operators.  **Security**  Spark enhances security compared to Hadoop by utilizing HDFS ACLs (Access Control Lists) and file-level permissions. Moreover, Spark can run on YARN with support for Kerberos authentication, further bolstering its security capabilities. | **15** | **L3** | **4** | **5** |

**\*Program Indicators are available separately for Computer Science and Engineering in AICTE examination reforms policy.**

**Course Outcome (CO) and Bloom’s level (BL) Coverage in Questions**



**Approved by the Audit Professor/Course Coordinator**