Internal Assessment Test 2 – DEC 2023 scheme and solution

Sub:	Big Data Analytics					Sub Code:	18CS72	Branch:	CSE	
Date:	6/12/2023	Duration:	90 mins	Max Marks:	50	Sem / Sec:	7 –A/B/C		OBE	

	Answer any FIVE FULL Questions	MARKS	CO	RBT
1	i) What is the work of Record Reader in map reduce process? (2 marks) ii) The output of map is in ---------- form. Output of map is (1 marks) iii) What is the difference between reducer and combiner? (2 marks) iv) Can Map Reduce process have 0 reducers? (Option: Yes or No) If so are combiners are necessary to be added? Justify. (1+ 2 marks) v) If the file size is 1024 MB and block size is 64 MB, how many mappers are needed to process the job? (2 marks)	[10]	CO 4	L2
2 a	An e-commerce website wants to track user interactions and transactions in real-time. They need a database that can handle rapid writes and queries for customer behaviors, purchase history, and product recommendations. Which NOSQL database would you recommend to meet the real-time tracking and querying requirements of the e-commerce website? Justify your answer Type of database: 1 Marks ,Justification : 4 Marks	[5]	CO 3	L4
2 b	A ride-sharing service needs to store and retrieve geospatial data, such as the real-time location of drivers and passengers, as well as route information. Efficient geospatial indexing and querying are crucial. Considering the importance of geospatial data and efficient querying, which type of NoSQL database would be the most appropriate for the ride-sharing service? Elaborate on your choice. Type of database: 1 Marks , Justification : 4 Marks	[5]	CO 3	L4
3	Explain ORC file format with features by taking automated chocolate vending machine (ACVM) example. Explain Parquet file format with suitable diagram. ORC file format and ACVM structure - 3+2 marks Parquet file format and diagram – 4+1 marks	[10]	CO 2, CO 3	L3
4	Explain MapReduce execution steps by considering wordcount example. MapReduce with all steps explanation – 6 marks Diagram – 2 marks Example -2 marks	[10]	CO 4	L3
5	Write Cassandra queries for i) Create a table Productlnfo in the keyspace lego, with primary key field Productld (4 marks) ii) Display the details of a table Productlnfo (3 marks) iii) Add a column dateOfManufacturing in the table Productlnfo (3 marks)	[10]	CO 3	L3
6	Explain ACID and BASE properties. Explain CAP theorum along with its importance in distributed architecture. ACID and BASE properties – 6 marks CAP theorem and its importance- 4 marks	[10]	CO 3	L2

Solution:

1.

i) What is the work of Record Reader in map reduce process? (2 marks)

- Converting record to key-value pair

ii) The output of map is in ---------- form. Output of map is (1 marks)

- Key-value pair

iii) What is the difference between reducer and combiner? (2 marks)

- Reducer works on mapper results from different nodes
- Combiners combines output of mapper from local node

iv) Can Map Reduce process have 0 reducers? (Option: Yes or No)

- yes

If so are combiners are necessary to be added? Justify. (1+ 2 marks)

Yes to combine results from the output of mapper

Combiners are optional components in the MapReduce framework that can be used to perform local aggregation on the output of the mappers before sending the data to the reducers. The use of combiners is not directly related to the number of reducers but rather to the efficiency of data transfer between mappers and reducers.

v) If the file size is 1024 MB and block size is 64 MB, how many mappers are needed to process the job? (2 marks)

- 1024/64= 16 mappers

==========

2.

An e-commerce website wants to track user interactions and transactions in real-time. They need a database that can handle rapid writes and queries for customer behaviors, purchase history, and product recommendations.

Which NOSQL database would you recommend to meet the real-time tracking and querying requirements of the e-commerce website? Justify your answer

Type of database: 1 Marks ,Justification : 4 Marks

Ans:

Cassandra- columnar database

Distributed and Scalable: Apache Cassandra is designed to be distributed and highly scalable. It can handle large amounts of data and traffic by distributing data across multiple nodes in a cluster.

As the e-commerce website grows, additional nodes can be added to the Cassandra cluster to scale both storage and throughput.

High Write Throughput: Cassandra is optimized for high write throughput, making it well-suited for real-time data tracking. It employs a distributed architecture with a peer-to-peer model, allowing it to handle a large number of concurrent writes across the cluster.

This is crucial for capturing and processing real-time user interactions and transactions.

No Single Point of Failure: Cassandra is fault-tolerant and has no single point of failure. Data is replicated across nodes, providing high availability and ensuring that the system remains operational even if some nodes fail.

This feature is critical for maintaining the reliability of the tracking and querying systems in an e-commerce setting.

Flexible Schema Design: Cassandra offers a flexible schema design, allowing the e-commerce website to adapt to changes in data requirements over time.
This flexibility is valuable in scenarios where the data model may evolve, such as adding new types of interactions or adjusting the schema for product recommendations.
Low Latency Queries: Cassandra supports low-latency queries, making it suitable for real-time querying of customer behaviors, purchase history, and product recommendations.
Users can quickly retrieve the information they need for personalized experiences on the e-commerce platform.
Time-Series Data Support: Cassandra is well-suited for time-series data, making it appropriate for tracking temporal events such as user interactions and transactions. Time-series data is a common requirement in e-commerce analytics for understanding user behavior patterns and trends.

==========

2b. A ride-sharing service needs to store and retrieve geospatial data, such as the real-time location of drivers and passengers, as well as route information. Efficient geospatial indexing and querying are crucial.
Considering the importance of geospatial data and efficient querying, which type of NoSQL database would be the most appropriate for the ride-sharing service? Elaborate on your choice.
Type of database: 1 Marks, Justification : 4 Marks
Ans:
MongoDB: columnar database
2D and 3D Geospatial Queries: MongoDB supports both 2D and 3D geospatial queries, making it versatile for various use cases. In a ride-sharing service, where location tracking involves latitude, longitude, and potentially altitude, the ability to handle 3D geospatial data is valuable.
Flexible Schema: MongoDB's flexible schema allows the storage of diverse geospatial data types. This flexibility is useful when dealing with different types of location-related information, such as routes, points of interest, and dynamic movements.
Integration with Application Logic: MongoDB's rich query language allows for the integration of geospatial queries with other application logic. This is important for a ride-sharing service where location-based queries are often combined with additional criteria, such as driver availability or passenger preferences.
Community and Ecosystem: MongoDB has a large and active community, providing ample resources and support for developers working with geospatial data.
The MongoDB ecosystem includes tools like MongoDB Compass, which offers a graphical interface for visualizing and interacting with geospatial data.
==========
3.
Explain ORC file format with features by taking automated chocolate vending machine (ACVM) example.
Explain Parquet file format with suitable diagram.
Ans:

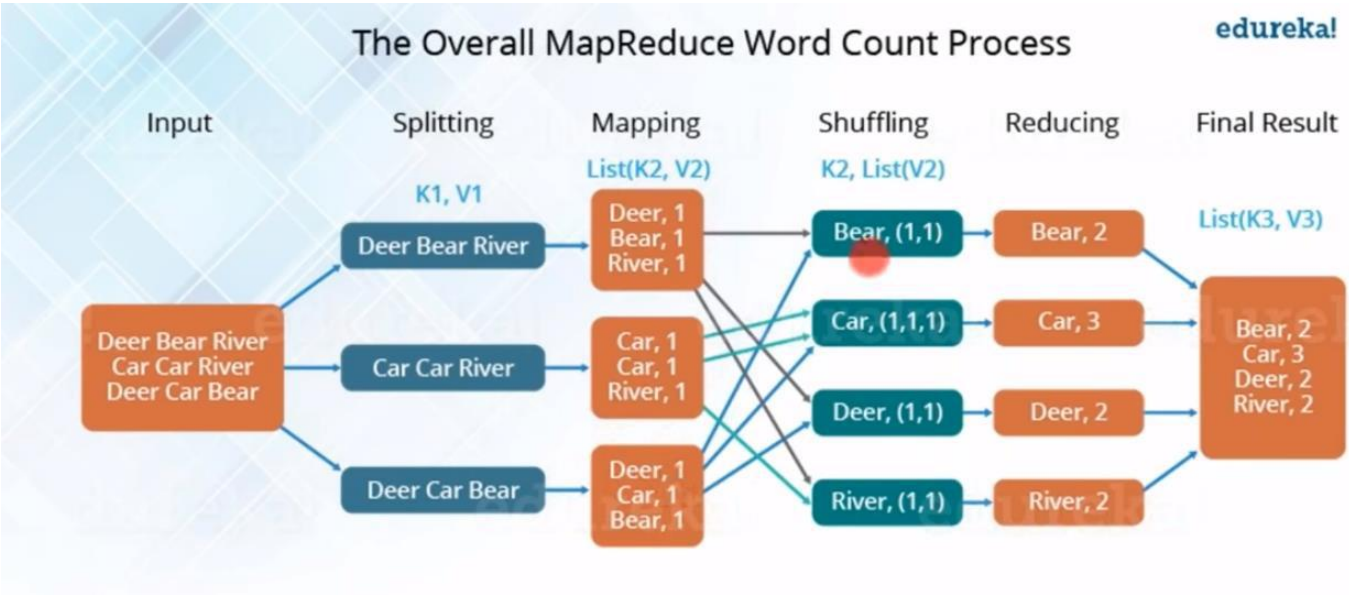
- **An ORC (Optimized Row Columnar)** file consists of row-group data called stripes.
- ORC enables concurrent reads of the same file using separate RecordReaders. Metadata store uses Protocol Buffers for addition and removal of fields.
- ORC is an intelligent Big Data file format for HDFS and Hive.

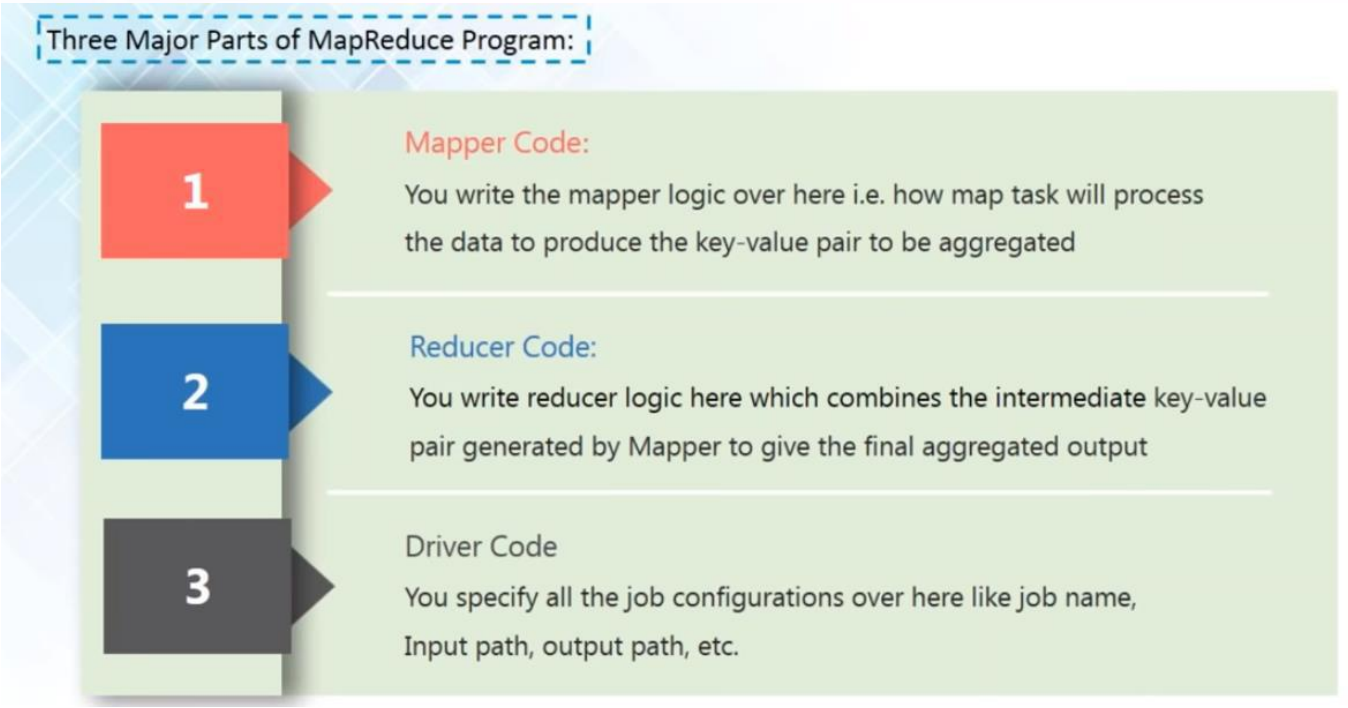
- An ORC file stores a collections of rows as a row-group. Each row-group data store in columnar format. This enables parallel processing of multiple row-groups in an HDFS cluster.
- An ORC file consists of a stripe the size of the file is by default 256 MB.
- Stripe consists of indexing (mapping) data in 8 columns, row-group columns data (contents) and stripe footer (metadata).
- An ORC has two sets of columns data instead of one column data in RC. One column is for each map or list size and other values which enable a query to decide skipping or reading of the mapped columns.
- A mapped column has contents required by the query. The columnar layout in each ORC file thus, optimizes for compression and enables skipping of data in columns. This reduces read and decompression load.
- Lightweight indexing is an ORC feature.
- Each index includes the aggregated values of minimum, maximum, sum and count using aggregation functions on the content columns.
- Therefore, contents column key for accessing the contents from a column consists of combination of row-group key, column mapping key, min, max, count (number) of column fields of the contents column.
- Table 3.5 gives the keys used to access or skip a contents column during querying. The keys are Stripe_ID, Index-column key, and contents-column name, min, max and count.
- The throughput increases due to skipping and reading of the required fields at contents-column key. Reading less number of ORC file content-columns reduces the workload on the NameNode.

Stripe_ID	Index Column 1				Index Column 2
	Index column 1 key 1				Index column 2 key 1
	Contents-Column name	Contents Minimum value	Contents Maximum value	Count (number) of content-column fields	
	...	...	...	...	
	...	...	...	...	
	Index column 1 key 2				Index column 2 key 2
	Column-name	Minimum value	Maximum value	Count of number of column fields	
	...	...	...	...	
	...	...	...	...	

==========

4. Explain MapReduce execution steps by considering wordcount example.





Partitioning

- The Partitioner does the partitioning The partitions are the semi mappers in MapReduce Partitioner is an optional class
- MapReduce driver class can specify the Partitioner A partition processes the output of map tasks before submitting it to Reducer tasks
- Partitioner function executes on each machine that performs a map task
- Partitioner is an optimization in MapReduce that allows local partitioning before reduce task phase Typically, the same codes implement the Partitioner Combiner as well as reduce() functions
- Functions for Partitioner and sorting functions are at the mapping node The main function of a Partitioner is to split the map output records with

the same key

Combiners

Combiners are semi reducers in MapReduce . Combiner is an optional class. MapReduce driver class can specify the combiner. The combiner() executes on each machine that performs a map task.

Combiners optimize MapReduce task that locally aggregates before the shuffle and sort phase. Typically, the same codes implement both the combiner and the reduce functions

The main function of a Combiner is to consolidate the map output records with the same key. The output (key value collection) of the combiner transfers over the network to the Reducer task as input

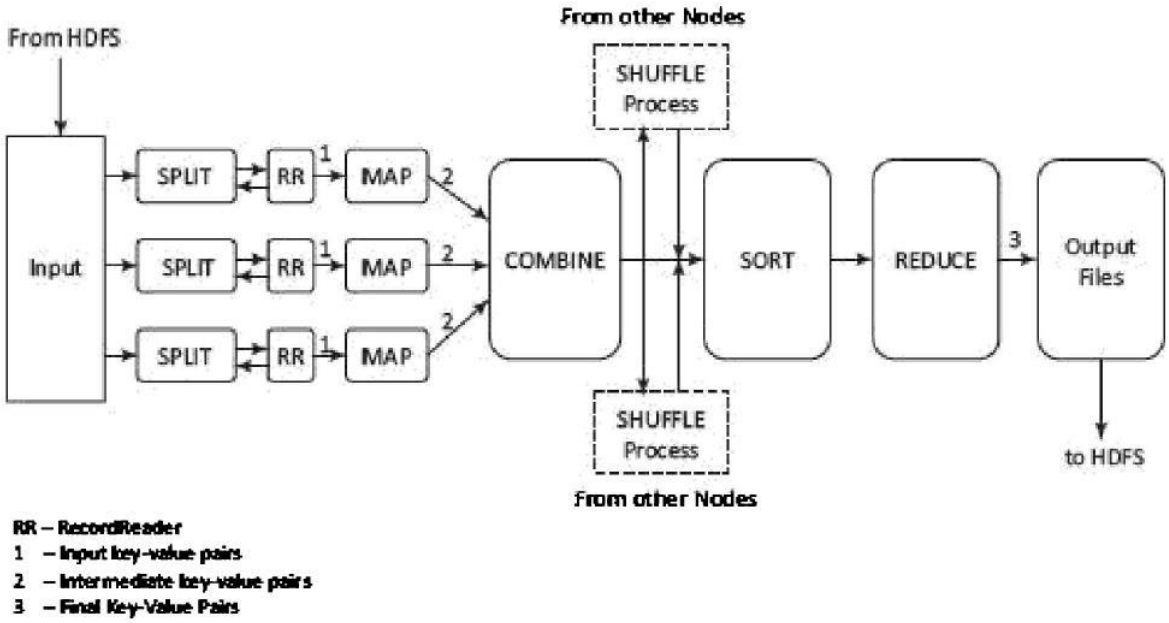


Figure 4.6 MapReduce execution steps

==========

5. Write Cassandra queries for

i) Create a table Productlnfo in the keyspace lego, with primary key field Productld (4 marks)

Use lego; Create table Productinfo(Productid int primary key, ProductType text);

ii) Display the details of a table Productlnfo (3 marks)

DESCRIBE TABLE Productinfo;

iii) Add a column dateOfManufacturing in the table Productlnfo (3 marks)

ALTER TABLE timestamp; ProductInfo add dateOfManufacturing

==========

6. Explain ACID and BASE properties. Explain CAP theorum along with its importance in

distributed architecture.

ACID

- *Atomicity* of transaction means all operations in the transaction must complete, and if interrupted, then must be undone (rolled back). Atomicity means both transactions should be completed, else undone if interrupted in between.
- Example
- *Consistency in transactions* means that a transaction must maintain the integrity constraint, and follow the consistency principle.
- Example
- *Isolation* of transactions means two transactions of the database must be isolated from each other and done separately
- *Durability* means a transaction must persist once completed.

BASE

CAP theorem

Any two properties must be satisfied

Consistency means all copies have the same value like in traditional DBs.

Availability means at least one copy is available in case a partition becomes inactive or fails. For example, in web applications, the other copy in the other partition is available.

Partition means parts which are active but may not cooperate (share) as in distributed DBs.

1. Consistency

   All nodes observe the same data at the same time. Therefore, the operations in one partition of the database should reflect in other related partitions in case of distributed database.

   Operations, which change the sales data from a specific showroom in a table should also reflect in changes in related tables which are using that sales data.

2. Availability

Availability means that during the transactions, the field values must be available in other partitions of the database so that each request receives a response on success as well as failure. (Failure causes the response to request from the replicate of data).

Distributed databases require transparency between one another.

Network failure may lead to data unavailability in a certain partition in case of no replication.

Replication ensures availability.

3. Partition

Partition means division of a large database into different databases without affecting the operations on them by adopting specified procedures

: