Roll No: | | | | | | | | | | | |

PRESIDENCY UNIVERSITY

GAIN MORE KNOWLEDGE
REACH GREATER HEIGHTS

Presidency University Act, 2013 of the Karnataka Act No. 41 of 2013 | Established under Section 2(f) of UGC Act, 1956
Approved by AICTE, New Delhi | Approved By BCI
Bengaluru

Mid Term Examinations - October 2026

Date: 05/10/2026 **Time:** 11:45 AM - 01:15 PM

Course Code: CBD3401 **Course Name:** Distributed Computing with Hadoop

Semester: Seventh Semester **Max. Marks:** 50 **Weightage:** 50%

CO - Levels	CO1	CO2
Marks	44	46

PART A: Answer Following Questions. 10 M

Q.N.	Questions	M	CO	BT
1	Fault tolerance is the ability of a distributed system to continue functioning even when one or more components fail. It can be achieved using techniques such as data replication, redundancy, and failure recovery mechanisms.	2	CO1	BT2
2	Hadoop is an open-source framework for distributed storage and processing of large datasets across clusters of computers. Two components are: HDFS – Hadoop Distributed File System for distributed data storage. MapReduce – Framework for distributed processing of large datasets.	2	CO1	BT1
3	Data replication is the process of storing multiple copies of each data block on different DataNodes. It is required to provide fault tolerance and data availability when a DataNode fails.	2	CO2	BT2
4	Data locality is the principle of moving computation closer to where the data is stored, rather than transferring large amounts of data across the network. It reduces network traffic and improves processing performance.	2	CO2	BT1
5	Mapper: Processes input data and generates intermediate key-value pairs. Reducer: Processes the intermediate key-value pairs and produces the final output.	2	CO2	BT1

PART B : Answer Any 1 Following Questions. 10 M

Q.N.	Questions	M	CO	BT
6	Scalability is the ability of a distributed system to handle an increasing workload by increasing its available resources. For example, if the number of users accessing an application increases significantly, the system should be able to accommodate the additional workload. Types of Scalability 1. Horizontal Scalability Horizontal scalability, also called scale-out, involves adding more machines or nodes to the system. Example: A Hadoop cluster initially contains 5 DataNodes. If the dataset and workload increase, additional DataNodes can be added. Advantages: Supports large-scale workloads. Provides increased storage and processing capacity. Can improve fault tolerance. Suitable for distributed systems. 2. Vertical Scalability Vertical scalability, also called scale-up, involves increasing the resources of an existing machine. Resources may include: CPU RAM Storage Network capacity Example: Increasing a server from 16 GB RAM to 64 GB RAM.	10	CO1	BT2

ANSWER KEY

PTO (Please Turn Over)

7	Fault tolerance is the ability of a distributed system to continue providing services even when one or more components fail. Failures may occur because of: Hardware failure Network failure Software errors Power failure Node failure Storage failure Fault-Tolerance Techniques 1. Replication Multiple copies of data or services are maintained. Example: HDFS stores multiple replicas of a data block on different DataNodes. 2. Redundancy Additional resources are maintained so that another resource can take over when a component fails. 3. Checkpointing The state of a computation is periodically saved. If a failure occurs, processing can restart from the most recent checkpoint. 4. Failure Detection The system continuously monitors nodes to identify failed or unavailable components. 5. Automatic Recovery When a failure is detected, the system automatically recovers or reallocates the task. 6. Load Balancing Workloads are distributed across available nodes to prevent individual nodes from becoming overloaded. Example in Hadoop Suppose a file is divided into blocks and each block has three replicas: Block A → DataNode 1, DataNode 2, DataNode 3 If DataNode 1 fails, the system can access Block A from DataNode 2 or DataNode 3. Benefits Improved availability Improved reliability Reduced data loss Continuous service Better recovery from failures	10	CO1	BT2

PART C: Answer Any 1 Following Questions. 10 M				
Q.N.	**Questions**	**M**	**CO**	**BT**
8	A Distributed File System (DFS) is a file system in which data is stored across multiple networked computers while providing users with a unified interface for accessing files. A large file can be divided into smaller blocks and distributed across multiple storage nodes. Characteristics 1. Distributed Storage Files are stored across multiple machines. 2. Scalability Storage capacity can be increased by adding additional nodes. 3. Fault Tolerance Replication and recovery mechanisms protect against node failures. 4. Data Availability Data remains accessible even when some nodes become unavailable. 5. Data Locality Computation can be performed close to where data is stored. 6. Transparency Users can access distributed files without needing to know their physical storage location. Advantages Large-scale storage High availability Fault tolerance Improved scalability Efficient utilization of storage resources Support for big-data applications Basic Architecture A typical DFS may contain: Client → Metadata Server → Storage Nodes The metadata component maintains information about files and their locations, while storage nodes contain the actual data. Example: HDFS HDFS uses: NameNode for metadata management. DataNodes for storing actual data blocks. Requirements of a Distributed File System Reliability Scalability Fault tolerance Security Efficient data access Data consistency Recovery mechanisms	10	CO1	BT2
9	Hadoop is an open-source framework designed for storing and processing very large datasets across a cluster of computers. Hadoop Architecture The major layers of Hadoop architecture are: 1. HDFS Hadoop Distributed File System provides distributed storage. It consists primarily of: NameNode: Maintains metadata. DataNode: Stores data blocks. 2. YARN Yet Another Resource Negotiator manages cluster resources and schedules jobs. Major components include: ResourceManager NodeManager ApplicationMaster 3. MapReduce MapReduce is a distributed processing framework. Its major stages include: Input Mapper Shuffle and Sort Reducer Output Hadoop Ecosystem The Hadoop ecosystem contains several technologies that support storage, processing, management, and analysis of big data. Hadoop Cluster A typical Hadoop cluster contains: Client → Master Services → Worker Nodes The worker nodes provide storage and processing resources, while master services coordinate cluster operations. Advantages of Hadoop Distributed storage Horizontal scalability Fault tolerance Cost-effective commodity hardware Parallel data processing Suitable for large datasets Applications Big-data analytics Log analysis Recommendation systems Fraud detection Search engines Machine learning Social media analytics	10	CO1	BT2

PART D: Answer Any 1 Following Questions. 10 M

ANSWER KEY

Q.N.	Questions	M	CO	BT
10	Data Locality Data locality refers to executing computation as close as possible to the location where the required data is stored. Types of Data Locality 1. Node Locality The computation runs on the same node where the required data block is stored. 2. Rack Locality If node-local execution is not possible, Hadoop attempts to execute the task on another node in the same rack. 3. Off-Rack Locality If neither node-local nor rack-local execution is possible, the task may execute on a node in another rack. Benefits of Data Locality Reduces network traffic. Improves processing performance. Reduces network congestion. Increases overall cluster efficiency. Fault Recovery Fault recovery refers to restoring normal operation after a component failure. Common Failures DataNode failure Task failure Network failure Disk failure DataNode Failure Recovery NameNode receives periodic heartbeat messages. If a DataNode stops sending heartbeats, it is considered unavailable. The NameNode identifies blocks affected by the failure. Existing replicas on other DataNodes are used. Missing replicas are recreated on healthy DataNodes. Task Failure Recovery If a processing task fails, Hadoop can execute the task again on another available node. Importance Data locality and fault recovery together provide: High performance High availability Fault tolerance Efficient resource utilization	10	CO2	BT2
11	MapReduce is a distributed programming model used to process large datasets in parallel across a Hadoop cluster. 1. Input Splitting The input data is divided into logical splits that can be processed independently. 2. Mapper The Mapper processes input records and generates intermediate key-value pairs. 3. Combiner A Combiner may perform local aggregation before data is transferred to the Reducer. 4. Partitioner The Partitioner determines which Reducer should receive each intermediate key. 5. Shuffle and Sort Intermediate data is transferred to Reducers and grouped according to keys. 6. Reducer The Reducer processes each key and its associated values. 7. Output The final results are stored in the distributed file system. Advantages Parallel processing Scalability Fault tolerance Data locality Suitable for large datasets	10	CO2	BT2

PART E: Answer Any 1 Following Questions. 10 M

Q.N.	Questions	M	CO	BT
12	MapReduce uses several components to transform and aggregate large datasets. 1. Mapper The Mapper converts input records into intermediate key-value pairs. 2. Combiner The Combiner performs local aggregation of Mapper output. 3. Partitioner The Partitioner determines which Reducer should process each key. 4. Shuffle and Sort The intermediate results are transferred to Reducers and grouped by key. 5. Reducer The Reducer aggregates the values.	10	CO2	BT2

ANSWER KEY

13	A MapReduce job execution workflow describes how an input dataset is transformed into final output through multiple stages. Step 1: Job Submission The client submits the MapReduce job along with: Input location Output location Mapper Reducer Configuration parameters Step 2: Input Splitting The input file is divided into logical input splits. Step 3: RecordReader The RecordReader converts input data into key-value pairs that are supplied to the Mapper. Step 4: Mapper Execution For Word Count: Step 5: Combiner The Combiner may locally aggregate values: Step 6: Partitioning The Partitioner determines the Reducer responsible for each key. Step 7: Shuffle and Sort Intermediate data is transferred to Reducers and grouped by key: Step 8: Reducer Execution The Reducer sums the values: Step 9: Output The final results are stored in HDFS. Fault Tolerance During Execution If a task fails, Hadoop can reschedule the task on another available node. This allows the overall job to continue despite individual task failures. Advantages of MapReduce Workflow Parallel execution Automatic task distribution Fault recovery Data locality Scalability Efficient processing of large datasets	10	CO2	BT2

ANSWER KEY