Student Name:Hassan Akram

Student ID: 221980043

SubmitED to Sir Zohoib

Course: Big Data

Part A: Conceptual (20 Marks)

1. Roles of Hadoop Components

Hadoop's architecture is made up of several critical components, each playing a unique role in ensuring efficient and reliable data management. The **NameNode** serves as the central authority by maintaining metadata information such as filenames, block locations, and access permissions. It is responsible for tracking where each piece of data is stored across the cluster. Complementing this, **DataNodes** are responsible for storing the actual content—whether it be text, images, or geographical data—and regularly send status updates to the NameNode. To assist in metadata management, the **Secondary NameNode** periodically merges the file system image (FsImage) with the transaction log (EditLog), which helps prevent excessive memory use and supports recovery processes. The system also implements **rack awareness**, which ensures data replicas are stored on different physical racks, increasing fault tolerance by minimizing the risk of data loss in case an entire rack fails. Additionally, **checkpointing** is employed to create a fresh FsImage from the merged logs, reducing the workload on the NameNode and improving system stability.

2. Five Key Reasons for Using HDFS

The Hadoop Distributed File System (HDFS) offers several significant advantages that make it suitable for managing large volumes of data. First, it is **highly scalable**, allowing organizations to seamlessly expand storage capacity by adding more DataNodes as data grows. Secondly, **fault tolerance** is built into the system through data replication, ensuring that even if some nodes fail, the data remains accessible. Third, HDFS is designed for **high throughput**, enabling fast processing of large datasets, which is essential for big data analytics. Another advantage is its **cost-effectiveness**, as it can run on low-cost, commodity hardware, reducing infrastructure expenses. Lastly, HDFS embraces the principle of **data locality**, which means it brings computation closer to where the data resides, thereby minimizing data transfer and improving processing efficiency.

Part B: Analytical Simulation & Numerical Problem Solving

1. Q1. HDFS Block Planning and Replicationsa) Number of HDFS Block
   - Demographic Text Files (100 TB): 409,600 block
   - Household Images (60 TB): 245,760 block
   - Geo-coordinates (20 TB): 81,920 block

b) Total Number of Block After Replication
- Total original block: 737,280
- After replication: 2,211,840 block

c) Total Storage Required
- 565,831,680 MB → 539.7 TB

2. Q2. Name Node Metadata Analysis

a) Metadata Before Replication: 175.7 MB
b) Metadata After Replication: 527.4 MB
c) RAM Check: Fits comfortably in Name Node RAM (32,768 MB)

3. Q3. Cluster Size Planning

a) Data Nodes for Census Data: 135
b) With 1 Node Failure: 136
c) IoT Data (120 TB): 90 Data Nodes

Q4. Advanced Scenario Simulation

a) **Under-Replication Event:** A failure involving seven DataNodes going offline results in approximately 28 TB of data being under-replicated across the cluster. This scenario poses a risk to data availability and resilience.

b) **Recovery Process:** Once maintenance is completed and nodes are restored or replaced, the **NameNode automatically initiates the re-replication process**. It identifies the missing replicas and instructs healthy DataNodes to replicate the affected data block, restoring the desired replication factor.

c) **MapReduce Response Mechanisms:** During the disruption, **MapReduce employs several strategies to maintain job progress**. These include **speculative execution**, where duplicate tasks are launched on different nodes to mitigate slowdowns; **dynamic task reassignment**, which reallocates failed or delayed tasks to active nodes; and **utilization of remaining replicas**, ensuring computations proceed using the available copies of the data block.