Big Data Analytics Assignment 1

Submitted By: Meer Hamza 221980020

June 2025

1 Part A: Conceptual

1.1 NameNode

NameNode, which is also known as MasterNode, manages the HDFS namespace and metadata, including the file-to-block mappings of the 180 TB census data. It tracks where each block, e.g. text, image, or CSV is stored across DataNodes but does not store the actual data itself.

1.2 DataNode

DataNodes store the actual data blocks of the 180 TB dataset (100 TB text, 60 TB images, 20 TB CSV) and handle read/write operations. Each DataNode has 4 TB of usable space and stores data blocks (256 MB each) with a replication factor of 3, ensuring fault tolerance. For example, if one DataNode fails, the data is still accessible from the other two replicas. DataNodes report block information to the NameNode and serve data to clients as directed.

1.3 Secondary NameNode

The Secondary NameNode periodically merges the Edit Logs of the NameNode and the Image of the File System (fsimage) to reduce the risk of loss of metadata and ensure faster recovery in the event of failure of the NameNode, supporting the 32 GB RAM limit.

1.4 Rack Awareness

Rack Awareness optimizes data placement and fault tolerance in the physical racks of the cluster. For Census data, HDFS ensures that the three replicas of each 256 MB block are distributed across different racks (e.g. two replicas on one rack, one on another). This ensures that if an entire rack fails, at least one replica remains accessible, supporting the requirement for fault tolerance against at least one DataNode failure.

1.5 Checkpointing

Checkpointing, performed by the Secondary NameNode, involves periodically consolidating the NameNode's edit logs into the fsimage. For the 180 TB dataset, this process ensures the NameNode can efficiently manage metadata within its 32 GB RAM by keeping the edit log size manageable.

1.6 Five Key Reasons why HDFS is preffered over tradi tional file systems

- Scalability: Easily handles petabytes of data by adding new DataNodes, ideal for future growth beyond 180 TB.
- Fault Tolerance: Data is replicated with factor of 3, this ensures that no data will loss even if a DataNode fails.
- High Throughput: Optimized for large file reads and writes, suitable for massive census datasets.
- Cost Efficiency: Runs on commodity hardware, reducing infrastructure costs for nationwide digitization.
- Data Locality: HDFS moves computation to the data, minimizing net work overhead for processing the census data. Traditional file systems often require data transfer to processing units, which is inefficient for such large datasets

2 Part B: Numerical Problem Solving 2.1

HDFS Block Planning and Replication

Given That:

HDFS Block Size = 256 MB
Replicate Factor = 3

2.1.1 Calculating how many blocks are needed

Data:

Demographic Text Files (100 TB)
Household Images (60 TB)
Geo-Coordinates (20 TB)

- Demographic Text Files:

1 TB = 1,048,576 MB

$$100 \times 1{,}048{,}576$$

$$256 = 104{,}857{,}600$$

$$256 \qquad 2 \qquad \boxed{409{,}600}$$

locks

• Household Images:

256=62,914,560

60 × 1,048,576

256=

• Geo-Coordinates: 245,760 Blocks

20 × 1,048,576

256=20,971,520 81,920 locks

256

• Total Blocks Before

Replication: 409,600 + 737,280 Blocks

245,760 + 81,920 =

2.1.2 Total Number of Blocks After Replication:

737,280 ∗ 3 = Blocks

2.1.3 Total Storage Required:

2,211,840

Original Size = 180 TB, Replication Factor = 3

180 ∗ 3 TB

=

540

2.2 NameNode Metadata Analysis

Given Data:

Metadata for each block = 250 bytes
Blocks Before Replication = 737,280 (From Q-1)
Blocks After Replication = 2,211,840 (From Q-2)
NameNode RAM = 32 GB = 32 * 1024 * 1024 * 1024 =34,359,738,368

bytes 2.2.1 Memory Required Before Replication:

737,280 blocks × 250 bytes/block
=

184,320,000 bytes ≈ 175.78 MB

2.2.2 Memory Required After Replication:

2 ,

, 8

2 4

1 0

1 b

l	5
o	0
c	b
k	y
s	t
×	e
2	s

/ k

b =

l 3

o 552,960,000 bytes ≈ 527.34 MB

c

2.2.3 Can the 32 GB RAM NameNode handle it?: •

Memory Used After Replication ≈ 527.34 MB

• Available RAM = 32 GB ≈ 32,768 MB

Yes, the NameNode can easily handle the metadata, as only about 1.6% of the total RAM is required.

2.3 Cluster Planning Size

Given Data:

Total Dataset with replication = 540 TB
Replication factor = 3
additional data = 120 TB(also to be replicated)

2.3.1 How many DataNodes are needed to store the entire dataset with replication?

540 TB

DataNodes

4 TB per DataNode

=

135

2.3.2 How many additional DataNodes would be required to tolerate 1 node failure and still meet replication requirements?

To tolerate 1 failure without effecting replication, the system must maintain effective capacity of 540 TB even after losing 1 node.
Find number of nodes such that:

$$(N - 1) \times 4 \text{ TB} \geq 540 \text{ TB}$$

$$N \geq \frac{540}{4} + 4$$

$$4 = \frac{544}{4}$$

136 DataNodes

136 − 135 = 1

Additional DataNodes required to ensure data remain available after 1 node failure = 1

2.3.3 If your institution later adds 120 TB of IoT sensor data, how many more DataNodes will be needed?

Total new storage required(with replication):

120 TB × 3 = 360 TB

)ataNodes

360 TB

4 TB per Node

90 4

2.4 Advanced Scenario Simulation

Given Data:

- Total Storage with replication = 540 TB
- Replication factor = 3 (dropped to 2 during maintanance)
- Each node has 4 TB space = 4 × 1024 GB = 4096 GB
- Min Number of DataNodes = 136
- Blocks before Replication = 737,280

Blocks after replication = 2,211,840
5% DataNodes go offline unexpectedly during maintanance

2.4.1 If you had the minimum number of nodes from Q3, how much data becomes under-replicated?

Calculation:
Min Number of DataNodes = 136
Number of DataNodes offline = 5% of 136 = 6.8 = 7
Storage per DataNode = 4 TB
Total storage lost due to offline DataNodes = 7 × 4 TB = 28 TB
Each DataNode holds:

2, 211, 840 blocks

136 nodes ≈

So, 7 nodes fail mean:

16, 264 blocks/node

7 × 16, 264 ≈

113, 848 block replication lost

Blocks effected by offline DataNodes = 113,848
Each affected block loses 1 replica (from 3 to 2), so under-replicated data is the size of these blocks.

2.4.2 What strategy will Hadoop follow to restore replication once the nodes are back online?

When the 7 offline DataNodes come back online, Hadoop's NameNode notices that 113,848 blocks have only 2 copies instead of the needed 3. To fix this and get back to 3 copies per block, Hadoop follows a simple plan: First, the NameNode checks reports from DataNodes to find blocks with too few copies. It makes a list of these blocks, deciding which ones to fix first based on their importance and which DataNodes are free. Then, it picks DataNodes, including the ones just back online, to store new copies, making sure copies are spread across different racks so the data stays safe if a rack fails. Next, it tells DataNodes with existing copies to send about 27.78 TB of data (for the 113,848 blocks, each 256 MB) to the chosen DataNodes. After the data is copied, the DataNodes report back, and the NameNode updates its records to show that all blocks now have 3 copies again.

2.4.3 If this outage happens during a MapReduce job that depends on blocks from failed nodes, what challenges will occur, and how does Hadoop mitigate them?

challenges:

- Data unavailability

  The 7 offline DataNodes hold approximately 113,848 blocks. If a MapReduce job requires data from these blocks, tasks accessing those blocks will fail, as some blocks may have fewer than 3 replicas.

- Job delays

  Tasks dependent on unavailable blocks will stall, increasing job execution time or causing partial failures.

- Reduced fault tolerance

  With the replication factor dropped to 2 and 7 DataNodes offline, some blocks may become temporarily inaccessible if both remaining replicas are on offline nodes, though this is unlikely with proper rack-aware placement.

- Resource contention

  If Hadoop attempts to re-replicate data during the MapReduce job, it could compete for network and disk resources, slowing down the job.

Mitigation Strategies:

- Data locality

  Hadoop runs tasks on nodes with the needed data. If a block is unavailable, it tries another node with a copy. With 2 replicas, most blocks remain accessible unless both are on the 7 offline nodes.

- Task retries

  If a task fails due to missing data, Hadoop retries it up to 4 times on another node with a block copy.

- Rack awareness

  HDFS spreads replicas across racks, so even with 7 nodes offline, most blocks likely have at least one copy available.

- Fault tolerance design

  The cluster, designed for 1 node failure (137 nodes), can handle 7 offline nodes (5% of 136), keeping most data accessible.