Date:

HYUNDAI

Q3 The default block size in hadoop, which is 128 mb, may seem small compared to traditional file system.

1) Efficient Distributed Processing → Hadoop is designed for distributed storage and processing of large datasets across a cluster of commodity hardware. with smaller size, Hadoop can distribute and process data more efficiently across multiple nodes in the cluster.

2) Reduced Data Loss → In a distributed environment, the chances of node failure increases, and smaller block sizes reduces the amount of data loss in case of node failure.

3) Optimized Disk I/O → smaller block size can lead to better disk I/O performance, especially on systems with smaller disk block sizes. This is because smaller blocks align better with the underlying storage devices block size

Date:

Q2

Yes, you can change the replication factor and block size of files in Hadoop.

1) Replication factor :- The replication factor determines how many copies of each block of data are stored across the Hadoop cluster. By default, Hadoop replicates data three times. We can change this replication factor at the time of file creation using the

⇒ '-replication' in 'hadoop fs -put'

2) Block Size → Hadoop divides large files into smaller blocks, and the default block size in hadoop is 128 Mb.
We can specify diff block size when creating a file using
'Ddfs.block.size' in 'hadoop fs -D'.

Date

HYUNDAI

Q1 Hadoop is designed to be fault-tolerant primarily through two key mechanism

1) Data replication

2) Job recovery.

1) In hadoop's distributed file system, HDFS, data is stored across multiple nodes in a cluster. By default, Hadoop replicates data across three different nodes. This replication ensures that even if one or two nodes fail, the data remain accessible from other nodes where it's replicated.

2.) Hadoop's processing framework, Mapreduce, is fault-tolerant in its own right. When a node running a part of a MapReduce job fails, Hadoop re-executes the fail task on another available node. Hadoop keeps track of the progress of each task, so if a node fails before completing its assigned task, Hadoop can reassign the task to another node without starting the entire job from scratch.