answer – 7.1

1605023

We need to explain how data distribution skew and execution skew can be handled using virtual node partitioning.

Data Distribution Skew

i. virtual node partitioning basically distributes data distribution skew among several nodes. Also, a tuple can be shifted from a highly loaded node to less loaded node with mapping table. Thus, data distribution skew is handled.

Execution Skew

i. Data execution skew, that is, one node is targeted for processing a sequence of queries in a certain period of time. To distribute this load among several nodes, tuples can be redistributed among less loaded nodes with mapping table.

ii. Node replication using virtual node partitioning can also address the issue. A specific node facing a significant number of requests in a short time span can replicate its data among several other nodes to reduce load.

Date :

Ans.

answer — 7.2

1605023

We need to explain how a query is executed in parallel storage system with dynamically partitioned storage of a relation. Here, the query and the partitioning is on the same attribute.

In a distributed/parallel data storage system with virtual node partitioning, the partitioning table is usually in the master node or it is replicated and distributed among routers and client nodes for faster query processing.

This partitioning table helps corresponding node or router to do the mapping from virtual to real node for a particular query. Thus, partitioning table diverts queries in these nodes to appropriate real node containing data. Also, dynamic repartitioning consistently changes the partitioning table. So, these updates are replicated in all the instances of the table. Thus, query diversion is achived properly.

Ans.

answer – 8.1

1605023

Advantages and disadvantages of replication are discussed below –

advantages

i. availability of data when a node fails. Other nodes containing replicated data provides what client asks for.

ii. faster operation and response.

iii. availability of real-time data.

disadvantages

i. concurrent update of data is difficult.

ii. increasing storage size.

Ans.

answer - 8.2

1605023

We are using 64MB block size to store a file named "Yourld_HFDS" of size 10GB in Hadoop File System.

=> here, block number = 10GB ÷ 64MB = 160.

=> assuming, we have 10 nodes.

Then, 160 blocks can be distributed among these 10 nodes using Round-Robin partitioning as there is no scope of skewing.

=> now, each node contains 16 blocks.

- The NameNode will contain 160 blockID entries against given file name.

=> each datanode contains 16 blocks of data and we have 10 DataNodes for storing blocks.

- also, 10 more DataNodes will contain the replica of these 160 blocks and another 10 more DataNodes will contain those replica too.

Date :

=> so, there will be 30 DataNodes in total to store blocks and ensure 3-levels of replication.

- we could have stored replicas in the same DataNode but then we could not have leveraged the parallel processing of distributed file system.

Ans.