****

Prashanth u

****

Assignment 2

Scenario Based questions:

Will the reducer work or not if you use "Limit 1" in any HiveQL query?

---No It will not work.

Suppose I have installed Apache Hive on top of my Hadoop cluster using default metastore configuration. Then, what will happen if we have multiple clients trying to access Hive at the same time?

---The default metastore configuration allows only one Hive session to be opened at a time for accessing the metastore. Therefore, if multiple clients try to access the metastore at the same time, they will get an error.

Suppose, I create a table that contains details of all the transactions done by the customers:

CREATE TABLE transaction_details (cust_id INT, amount FLOAT, month STRING, country STRING) ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' ;

Now, after inserting 50,000 records in this table, I want to know the total revenue generated for each month. But, Hive is taking too much time in processing this query. How will you solve this problem and list the steps that I will be taking in order to do so?

---We can use partioned methods to query specificqlly

How can you add a new partition for the month December in the above partitioned table?

---ALTER TABLE transaction_details ADD partition month ;

I am inserting data into a table based on partitions dynamically. But, I received an error – FAILED ERROR IN SEMANTIC ANALYSIS: Dynamic partition strict mode requires at least one static partition column. How will you remove this error?

---SET hive.exec.dynamic.partition = true;

---SET hive.exec.dynamic.partition.mode = nonstrict;

Suppose, I have a CSV file – 'sample.csv' present in '/temp' directory with the following entries:

id first_name last_name email gender ip_address

How will you consume this CSV file into the Hive warehouse using built-in SerDe?

---Here, we are using the OpenCSVSerde which is a built-in SerDe for processing CSV files. The separator character, quote character, and escape character are specified using the SERDEPROPERTIES clause.

Load the data from the CSV file into the table using the following command:

LOAD DATA LOCAL INPATH '/temp/sample.csv' OVERWRITE INTO TABLE sample;

Here, we are using the LOAD DATA command to load the data from the local file system. The OVERWRITE keyword is used to replace any existing data in the table.

Suppose, I have a lot of small CSV files present in the input directory in HDFS and I want to create a single Hive table corresponding to these files. The data in these files are in the format: {id, name, e-mail, country}. Now, as we know, Hadoop performance degrades when we use lots of small files.

So, how will you solve this problem where we want to create a single Hive table for lots of small files without degrading the performance of the system?

---Merge the small CSV files into larger files: We can merge the small CSV files present in the input directory into larger files using Hadoop File APIs or tools like Hadoop DistCp. This step reduces the number of files in the input directory and improves the performance of the system.

---Copy the merged files to a new directory: After merging the small CSV files, we can copy the merged files to a new directory in HDFS. This new directory will contain larger files, which can be used to create a Hive table.

---Create a Hive external table: We can create a Hive external table that maps to the new directory where the merged files are stored. The external table can be created using the following HiveQL command:

LOAD DATA LOCAL INPATH ‘Home/country/state/’

LOAD DATA INPATH 'hdfs://<new_directory>/*' INTO TABLE table_name;

OVERWRITE INTO TABLE address;

The following statement failed to execute. What can be the cause?

---Incorrect path: The path specified in the statement may be incorrect. It is possible that the file or directory specified in the path does not exist, or there is a typo in the path name. Make sure that the path is correct and points to the right file or directory.

---Improper syntax: The syntax of the LOAD DATA statement may be incorrect. The statement should have a table name after INTO TABLE clause. It is also important to ensure that the syntax of the statement is correct and there are no typos or errors in it.

---Special characters: The use of special characters in the path name can also cause the LOAD DATA statement to fail. Make sure to escape any special characters in the path name using backslashes or by enclosing the path in quotes.

Is it possible to add 100 nodes when we already have 100 nodes in Hive? If yes, how?

---Yes, it is possible to add 100 nodes when we already have 100 nodes in Hive, provided that the underlying Hadoop cluster has the capacity to support the additional nodes. Here are the high-level steps to add 100 nodes to a Hive cluster:

---Add new nodes to the Hadoop cluster: First, we need to add the new nodes to the underlying Hadoop cluster. This can be done by configuring the new nodes with Hadoop and connecting them to the existing Hadoop cluster.

---Configure Hadoop and Hive: Next, we need to configure Hadoop and Hive to recognize the new nodes. This involves updating configuration files such as hdfs-site.xml, core-site.xml, and hive-site.xml to include the new nodes in the cluster.

---Start Hadoop and Hive services: Once the new nodes are added and configured, we need to start the Hadoop and Hive services to make them operational. This involves starting the HDFS, YARN, and MapReduce daemons on the new nodes and verifying that they are running correctly.

---Rebalance the data: After adding the new nodes, we need to rebalance the data across the entire Hadoop cluster to ensure that the data is distributed evenly across all nodes. This can be done using the Hadoop balancer tool.

---Scale out Hive: Finally, we need to scale out Hive to take advantage of the additional nodes. This involves configuring Hive to use additional resources such as memory and CPU on the new nodes, and potentially adding new partitions to tables to take advantage of the additional storage capacity.