Session14 Assignment 1- answers

1. **When Hive is best suited and when is it not?**

Hive is best suited for data warehousing applications where data is stored, mined and reporting is done based on processing. Hive bridges the gap between the data warehouse applications based on RDBMS and Hadoop. When it becomes more difficult to write complex mapreduce

Hive is not suited for OLTP (Online Transaction Processing) applications. It’s not a complete database and does not allow record level insert/update/delete operation on tables. Hence not meant to be used for any system which needs interactive processing.

1. **When should one use Hive over MapReduce?**
2. When it becomes more complex to code MapReduce program in JAVA.
3. To reduce development cycle effort.
4. When ad-hoc analysis has to be done on the data.
5. For data analytics purpose.
6. **What is Hive metastore?**

Hive metastore is the central repository of HIVE metadata. HIVE organizes the data into tables and the details like table schemas, partition details etc – called metadata are stored in metastore.

Meta data are stored in a relational database of metastore and metastore provides access to Hive clients to this meta data via meta store service API.

There can be 3 types of metastores :-

1. An embedded metastore , which is default Derby database and has a limitation to only one Derby database to have access to files on disk anytime, meaning only one session is possible.
2. Local metastore , which is a standalone database allowing multiple sessions at a time. (MySQL is a popular standalone database)
3. Remote metastore , where one or more metastore servers run in separate processes to the Hive service
4. **How can Hive improve performance with orc file format tables?**

The *Optimized Row Columnar* (ORC) file format provides a highly efficient way to store Hive data. Compared with RCFile format, ORC file format has many advantages such as:

1. a single file as the output of each task, which reduces the NameNode's load
2. Hive type support including datetime, decimal, and the complex types (struct, list, map, and union)
3. light-weight indexes stored within the file - skips row groups that don't pass predicate filtering and seeks to a given row.
4. block-mode compression based on data type - run-length encoding for integer columns and

dictionary encoding for string columns

1. concurrent reads of the same file using separate RecordReaders.
2. ability to split files without scanning for markers
3. bound the amount of memory needed for reading or writing
4. metadata stored using Protocol Buffers, which allows addition and removal of fields
5. takes less time to access data and takes less space to store data.
6. ORC stores collections of rows in one file and within the collection the row data is stored in a columnar format. This allows parallel processing of row collections across a cluster. Each file with the columnar layout is optimized for compression and skipping of data/columns to reduce read and decompression load.

**5. What is thrift server and client, jdbc and odbc driver importance in hive?**

Thrift server & client, JDBC and ODBC are different mechanisms to connect from other application to HIVE as a server, hence they are referred to as HIVE clients.

1. Thrift server & client - enabling access from a range of clients (applications) written in different languages supporting Thrift.
2. JDBC driver - a Java application will connect to a Hive server running in a separate process.
3. ODBC driver - An ODBC driver allows applications that support the ODBC protocol (such as business intelligence software) to connect to Hive
4. **What is the importance of partition in hive?**

Partition in hive distributes large data sets horizontally. So that the data can be organized in a logical parts. It enhances the query performance. When we write the query to fetch the values from table , only the required partitions of the table are queried, which reduces the time taken by query to yield the result.

In non-partitioned tables, Hive would have to read all the files in a table’s data directory and subsequently apply filters on it. This is slow and expensive—especially in cases of large tables.

1. **What is the use of bucketing in hive?**

To overcome over-partitioning, we can use bucketing in which we can decompose our data into more manageable parts called buckets.

Bucketing is based on hashing function based on bucketing column mod by total number of buckets, which stores the records with same bucketing column in the same bucket.

Thus bucketing is useful in

1. Provide faster query response
2. efficient sampling, and thereby useful for testing and debugging.

**8. What is the difference between static partitioning and dynamic partitioning in hive?**

In static partitioning every partitioning needs to be backed with individual hive statement which is not feasible for large number of partitions as it will require writing of lot of hive statements.

In that scenario dynamic partitioning is suggested as we can create as many number of partitions with single hive statement.

Static partitioning needs to be applied when we know data (supposed to be inserted) belongs to which partition. And When you have large data stored in a table then Dynamic partition is suitable.

Dynamic Partition takes more time in loading data compared to static partition, however If you want to partition number of column but you don’t know how many columns then also dynamic partition is suitable.