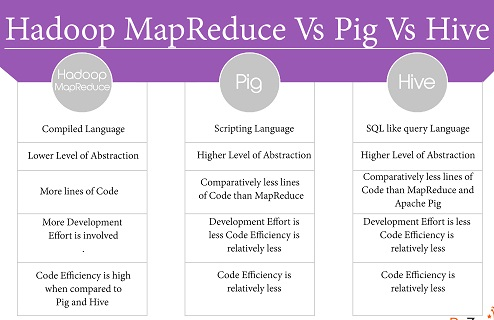
1. When Hive is best suited and when is it not?

* Pig-latin is data flow style, is more suitable for software engineer. While sql is more suitable for analytics person who are get used to sql. For complex task, for hive you have to manually to create temporary table to store intermediate data, but it is not necessary for pig.
* Pig-latin is suitable for complicated data structure( like small graph). There's a data structure in pig called DataBag which is a collection of Tuple. Sometimes you need to calculate metrics which involve multiple tuples ( there's a hidden link between tuples, in this case I would call it graph). In this case, it is very easy to write a UDF to calculate the metrics which involve multiple tuples. Of course it could be done in hive, but it is not so convenient as it is in pig.
* Writing UDF in pig much is easier than in Hive in my opinion.
* Pig has no metadata support, (or it is optional, in future it may integrate hcatalog). Hive has tables' metadata stored in database.
* You can debug pig script in local environment, but it would be hard for hive to do that. The reason is point 3. You need to set up hive metadata in your local environment, very time consuming.
* Hive is not a full database. The design constraints and limitations of Hadoop and HDFS impose limits on what Hive can do.
* Hive is most suited for data warehouse applications, where
  + Relatively static data is analyzed,
  + Fast response times are not required, and
  + When the data is not changing rapidly.

2. When should one use Hive over MapReduce?

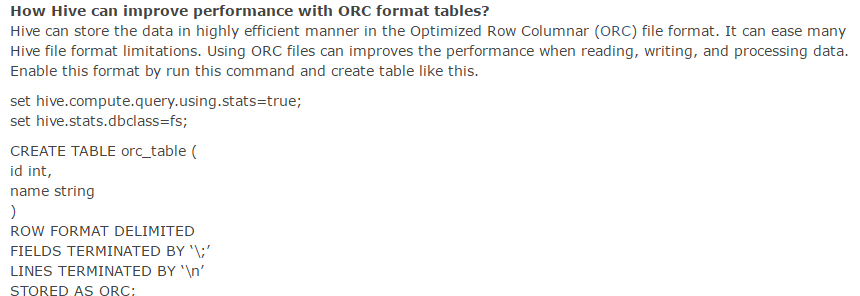
* When you need definite driver program control
* Whenever the job requires implementing a custom Partitioner
* If there already exists pre-defined library of Java Mappers or Reducers for a job
* If you require good amount of testability when combining lots of large data sets
* If the application demands legacy code requirements that command physical structure
* If the job requires optimization at a particular stage of processing by making the best use of tricks like in-mapper combining
* If the job has some tricky usage of distributed cache (replicated join), cross products, groupings or joins



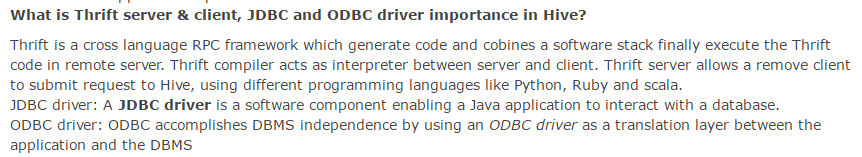
3. What is Hive metastore?

The **Hive metastore** service stores the metadata for **Hive** tables and partitions in a relational database, and provides clients (including **Hive**) access to this information via the **metastore**service API

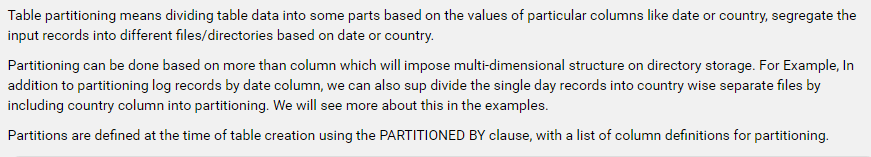
4. How can Hive improve performance with orc file format tables?

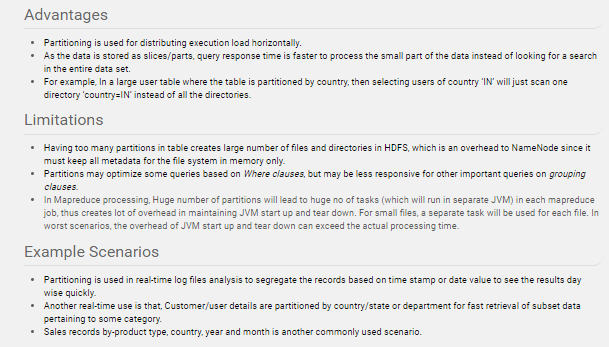


5. What is thrift server and client, jdbc and odbc driver importance in hive?

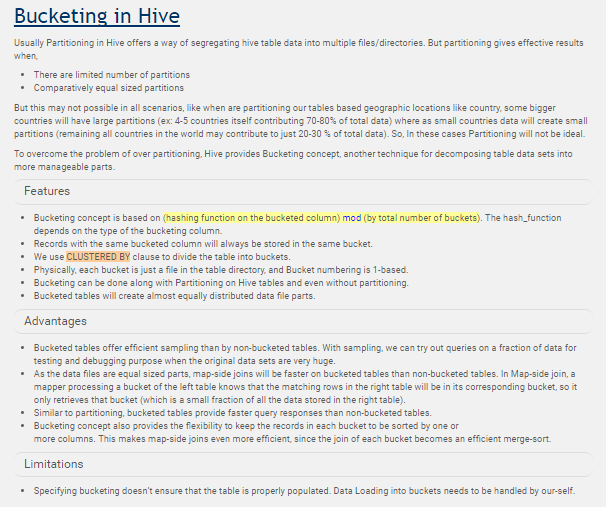


6.What is the importance of partition in hive?





7.What is the use of bucketing in hive?



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

partitioning in Hive is very useful to purne data during query to reduce query times.

Partitions are created when data is inserted into table. Depending on how you load data you would need partitions. Usually when loading files (big files) into Hive tables static partitions are preferred. That saves your time in loading data compared to dynamic partition. You "statically" add a partition in table and move the file into the partition of the table. Since the files are big they are usually generated in HDFS. You can get the partition column value form the filename, day of date etc without reading the whole big file.

Incase of dynamic partition whole big file i.e. every row of the data is read and data is partitioned through a MR job into the destination tables depending on certain field in file. So usually dynamic partition are useful when you are doing sort of a ETL flow in your data pipeline. e.g. you load a huge file through a move command into a Table X. then you run a inert query into a Table Y and partition data based on field in table X say day , country. You may want to further run a ETL step to partition the data in country partition in Table Y into a Table Z where data is partitioned based on cities for a particular country only. etc.

Thus depending on your end table or requirements for data and in what form data is produced at source you may choose static or dynamic partition.