Enable Sorted Bucketing in Hive

From the hive documents mostly we get to an impression as for grouping records, we go in for partitions and for sampling purposes, ie for evenly distributed records across multiple files we go in for buckets. But can we group records based on some columns/fields in buckets as well (individual files in buckets).

Concepts get clearer when we explain it through examples. So I’m taking the same route here.  Once with a hadoop assignment we did design a hadoop hybrid solution where the final output was on a hive partitioned table. This final output has to be consumed by an Oracle DWH for some legacy applications. The hand shake between hadoop and oracle team was they wanted ‘n’ files for each sub partition/folder and the files should have data grouped based on a few columns in the table (country and continent). If the files are grouped then the oracle load would be much efficient. How can we get the solution materialized?

1.       After hive operations do a map reduce on the final folders that would do the Group by

You do this by setting the number of reducers to ’n’ for n output files while running against each sub folder.It is really not a good solution because you have to run the map reduce for all sub partitions/folders which is definitely a performance glitch.

2.       Bucketing in hive

Using bucketing in hive for sub paritions. It is not plain bucketing but sorted bucketing. Normally we enable bucketing in hive during table creation as

CREATE EXTERNAL TABLE IF NOT EXISTS test\_table

(

  Id INT,name String

)

PARTITIONED BY (dt STRING,hour STRING)

CLUSTERED BY(country,continent) INTO n BUCKETS

ROW FORMAT DELIMITED FIELDS TERMINATED BY '|'

LOCATION '/home/test\_dir';

When we go into sorted bucketing/grouped bucketing our DDL would look like

CREATE EXTERNAL TABLE IF NOT EXISTS test\_table

(

Id INT,name String

)

PARTITIONED BY (dt STRING,hour STRING)

CLUSTERED BY(country,continent) SORTED BY(country,continent) INTO n BUCKETS

ROW FORMAT DELIMITED FIELDS TERMINATED BY '|'

LOCATION '/home/test\_dir';

Now to enforce bucketing while loading data into the table, we need to enable a hive parameter as follows

set hive.enforce.bucketing = true;

With this DDL our requirement would be satisfied. The n individual files within each sub partitions and the records would be grouped into n files based on country, continent. ie the a particular combination of country, continent would be present in only one file. Now if the question arises, which combination in which file? It is decided by the hash partitioning function. If you want control over that you need to write your custom hash partitioner and plug in the same into your hive session.

NOTE: When we use partitions data is stored under individual directories/sub directories in hdfs. But when we use buckets the records are stored as files with naming convention ranging from 0 to n-1.

NOTE: In partitioned tables when we issue a query only the required partitions are scanned, no need to specify any hints in your hive query. But for bucketed tables it is not the case, you need to hint your hive query if you want to scan some particular buckets else the whole set of files would be scanned. We hint the buckets using TABLESAMPLE clause in our hive query. For example in our example if we want to choose only the data from BUCKET 2

SELECT \* FROM test\_table TABLESAMPLE(2 OUT OF n BUCKETS)WHERE dt=’2011-10-11’ AND hr=’13’;

FAQ:

Bucking is basically used for sampling and not for queries that point to certain data group. Say you want to calculate the average of age(one column in table) and you not looking at exact average but an approximate one. In those cases bucketing is used . Also bucketing is used to optimize joins as well.   
  
Also when I mentioned '2 of n' buckets means, choose 2 buckets out of n. If I mention 'n on n' then whole data would be used as input.