CarbonData supports binary data type

Version	Changes	Owner	Date
0.1	Init doc for Supporting binary data type	Xubo	2019-4-10

Background :

Binary is basic data type and widely used in various scenarios. So it's better to support binary data type in CarbonData. Download data from S3 will be slow when dataset has lots of small binary data. The majority of application scenarios are related to storage small binary data type into CarbonData, which can avoid small binary files problem and speed up S3 access performance, also can decrease cost of accessing OBS by decreasing the number of calling S3 API. It also will easier to manage structure data and Unstructured data(binary) by storing them into CarbonData.

Goals:

1. Supporting write binary data type by Carbon Java SDK.[Formal]
2. Supporting read binary data type by Spark Carbon file format(carbon datasource) and CarbonSession.[Formal]
3. Supporting read binary data type by Carbon SDK
4. Supporting write binary by spark

Approach and Detail:

1.Supporting write binary data type by Carbon Java SDK [Formal]:

1.1 Java SDK needs support write data with specific data types, like int, double, byte[ ] data type, no need to convert all data type to string array. User read binary file as byte[], then SDK writes byte[] into binary column.

1.2 CarbonData doesn't compress binary column because binary usually is already compressed, like jpg format image. So no need to uncompress for binary column.

1.3 CarbonData stores binary as dimension.

1.4 Support configure page size for binary data type because binary data usually is big, such as 200k. Otherwise it will be very big for one blocklet (32000 rows).

2. Supporting read and manage binary data type by Spark Carbon file format(carbon DataSource) and CarbonSession.[Formal]

2.1 Supporting read binary data type from non-transaction table, read binary column and return as byte[]

2.2 Support create table with binary column, table property doesn't support sort_columns, dictionary, carbon.column.compressor, COLUMN_META_CACHE, RANGE_COLUMN for binary column

2.3 Support CTAS for binary

2.4 Support external table for binary

2.5 Support projection for binary column

2.6 Support show table, desc, ALTER TABLE for binary data type

2.7 Don't support PARTITION, filter, BUCKETCOLUMNS for binary

2.8 Support compaction for binary

2.9 datamap? Don’t support bloomfilter, lucene , timeseries datamap, support mv and priggregate in the future

2.10 CSDK / python SDK support binary in the future.

2.11 Support S3

3. Supporting read binary data type by Carbon SDK

3.1 Supporting read binary data type from non-transaction table, read binary column and return as byte[]

3.2 Supporting projection for binary column

3.3 Supporting S3

4. Supporting write binary by spark

4.1 Convert binary to String and storage in CSV, encode as Hex

4.2 Spark load CSV and convert string to binary, and storage in CarbonData. CarbonData internal will decode Hex to binary.

4.3 Supporting insert, update, delete for binary

4.4 Don’t support stream table.