Using Apache CarbonData

As an organisation, we were looking for a big data platform to execute the Analytics queries. While attending one of the Spark meetups we came across Apache CarbonData. Our interest in a platform capable of analytical query processing led us to get ourselves introduced to Apache CarbonData,so we began exploring Apache CarbonData.

First step was to to install and use the Apache CarbonData application. Installing, building and using Apache CarbonData is simple, even for those who are new to the Big Data Technology. We started with the Quick Start installation guide provided in the Apache CarbonData online documentation. The directions given in the quick start were pretty intuitive. Within no time we were able to do the entire setup and started executing some basic queries. The directions involved the following steps:-

1. Download and install Spark1.5.0 to Spark 1.6.2
2. Download and install Apache Thrift 0.9.3
3. Download Apache CarbonData code using the following command
   $ git clone https://github.com/apache/incubator-carbondata.git
4. Build the downloaded Apache CarbonData code using the following command
   $ mvn -DskipTests -Pspark-1.6 -Dspark.version=1.6.2 clean package

Creating the table was the first step. So, we used the Create table query to create a table with a mix of dimensions and measures. Next, we used a CSV file to load the data from. We used the Load query next. After the successful loading of data, we executed some basic select queries.

With the quick installation so easy and intuitive, we wanted to explore more. We thought of using the Apache CarbonData in a scenario similar to the real world. The next step was to do the Apache CarbonData setup on the Standalone Spark cluster. After following the online documentation and with the help from their community, we were able to complete the setup.

The Apache CarbonData Community is very active and were very quick and timely in clarifying and resolving our doubts during the entire phase.

As we are already familiar with Parquet and Hive, we tried to compare query performance between Apache CarbonData and Parquet. We tested the performance on two systems with the following configurations :-

- 500 GB data
- 300 columns
- Select query on 150 columns
- Cluster setup – 5 nodes, each node having 4 core and 8 GB

We loaded different sizes of data in the table – 10 GB, 100 GB, 500 GB, 5 TB, 300 columns. For each data, the query performance was much better with Apache CarbonData. With the increase in the data size, the difference in the query performance was even more visible.

Ease of loading the data was easily noticeable. The data can be loaded in the Carbon tables from a variety of sources. Apache CarbonData supports loading data from different files, from Parquet and Hive table, using DataFrames, and from JSON.

Another noticeable feature was the support for the varied data types. Along with the all basic data types, Apache CarbonData supports some complex ones as well, for e.g. Array, struct complex nested types.