COSC 6339 Big Data Analytics-Assignment3

-Sreekar Reddy Jammula (1422539)

Problem description:

The current assignment is to write the python scripts for Apache Spark. The tasks are divided into two parts as below:

1. **FileFormat Conversion-**To convert the given .csv file consisting of airline data into a parquet, json and sequence file
2. **Find Delayed Flights:** Develop the pyspark code to find the percentage of delayed flights per origin airport and measure the execution times for the csv ,parquet, json and sequence file formats on 5,10 and 15 executors.

Input Data Set: A large input data set consisting of flight data from year 2004 to 2008(about 28 million flights amounting to 3.5GB) present in the directory /cosc6339_s17/flightdata-full/ has been used for this experiments. The description of flight data is present at: http://stat-computing.org/dataexpo/2009/the-data.html

Outputs: All the outputs have been stored in the /bigd12/ Hadoop dfs directory

Solution Strategy:

The solution strategy is as follows:

1. Task1: For task1, the strategy is to split the data,transform it and then create a Dataframe out of it. These dataframes are written into a parquet or json files.For sequence file, the dataframes are not required and the data has to be split as a <key, value> with the key being empty. This data is then saved as a Sequence file.
2. Task 2: For task 2, the strategy is to write the SQL query that computed the percentage count directly. The query has to be written with the help of a spark SQL context object as shown in the code.

Programming Environment:

Apache Spark: Apache Spark allows us to easily create BigData applications on the cluster. It is generally considered to be faster than map reduce programming model and is very much useful for large scale applications.

1. It has API's defined for multiple languages and can working with a variety of filesystems and databases. However, for this experiment, we confine ourselves to the python API called PySpark and the Hadoop File System.
2. Spark treats the data in terms of RDD (Resilient Distributed Datasets) which are then manipulated using the transform and action operations.
3. Another main reason for using spark is that it offers powerful graph processing and Machine learning libraries.

PySpark: Pyspark is the python interface for Apache Spark. I

Executing the code:

To execute the the programs follow the following command :

```
spark-submit --master yarn --num-executors <<enter number of executors here>>   <<enter program name here>>
```

Example:

```
spark-submit --master yarn --num-executors 5   asst3_par.py
```

Description of files attached:

asst3_par.py: Contains code to convert CSV files into parquet files.

asst3_json.py: Contains code to convert CSV files into json files.

asst3_seq.py: Contains code to convert CSV files into sequence files.

asst3_par2.py: Contains code to find percentage of delayed flights for parquet files.

asst3_json2.py: Contains code to find percentage of delayed flights for json files.

asst3_seq2.py: Contains code to find percentage of delayed flights for sequence files.

asst3_csv2.py: Contains code to find percentage of delayed flights for csv files.

Results and Observations:

Task 1: The file sizes for different formats are measured and they are as follows:

Format	Parquet	JSON	Sequence
File Size	0.59GB	18.4GB	3.8GB

Observation: The Parquet file format occupies much less space than the original CSV format.
The Sequence file format occupies nearly the same space as CSV format
The JSON format however hold much larger space than CSV.

Task2-The execution times to find the percentage of delayed flights per origin are showed on a per origin basis in the below tables. Readings from trials which are inconsistent are ignored while taking the average and have been mentioned in the comments below the table. All readings are in seconds.

Number of executors =5

File Type	Trial 1	Trial 2	Trial 3	Average
CSV	600	415	491	453
Parquet	97	68	98	97.5
JSON	666	670	617	668
Sequence	444	291	290	290.5

Comments: For CSV, ignore trial 1
For Parquet, ignore the trial 2.
For JSON, ignore trial 3.
For Sequence, ignore trial 1.

Observation: Parquet files take the most amount of time to execute. JSON files take most amount of time.

Number of executors = 10:

File Type	Trial 1	Trial 2	Trial 3	Average
CSV	671	489	480	484.5
Parquet	135	140	67	137.5
JSON	540	556	546	543
Sequence	170	175	207	172.5

Comments: For CSV,

For Parquet, ignore trial 3.

For JSON, ignore trial 2.

For Sequence, ignore trial 3.

Observation: The parquet file take least amount of time and JSON take the most amount of time. The execution time increase for CSV and Parquet and decrease for JSON and Sequence files when compared to 5 reducers. Although the execution times are expected to decrease when increasing the number of executors by a small number, the results here may vary based on factors such as file size, availability of resources etc.,

Number of executors = 15

File Type	Trial 1	Trial 2	Trial 3	Average
CSV	595	600	755	597.5
Parquet	62	79	82	80.5
JSON	423	408	322	415.5
Sequence	218	288	315	253

Comments: For CSV, ignore trial 3.

For Parquet, ignore trial 1.

For JSON, ignore trial 3.

For Sequence, ignore trial 3.

Observation: Parquet files take the least amount of time and the JSON files take the highest amount of time. The execution times increase for CSV and Sequence and decrease for Parquet and JSON. In general, when the number of executors are increased continuously, a drop in the performance is observed because I/O overhead etc., However, the results here may depend upon factors such as size of files, availability of resources on clusters etc.

Resources Used:

These experiments have been performed on the Whale cluster which is under the supervision the parallel software technologies laboratory. The cluster has been accessed through a secured SSH connection from the MobaXterm SSH client.

Given below are the details about the cluster:

1. Total number of nodes: 57 (whale-001 to whale-0057)
2. Processor Details: 2.2 GHz quad-core AMD Opteron processor (consisting of 4 cores each)- 2 in number(8 cores total)
3. Gigabit Internet.
4. Storage: 20 TB Sun StorageTek 6140 array (/home shared with shark and crill clusters)
5. Network Interconnect:

   -144 port 4xInfiniBand DDR Voltaire Grid Director ISR 2012 switch (donation from TOTAL, shared with crill)

   -Two 48 port HP GE switch