New York City Taxi Trip Distance Analysis using BigData

Deepak Prasad
Masters of Data Science
RMIT
Melbourne, Australia
s3759108@student.rmit.edu.au

***Abstract*— This report provides an insight into the analysis of New York taxi dataset. Also, the scalability aspects in terms of data size has also been examined. A custom algorithm is used to determine the average distances travelled by customers during various phases of time. These insights can be used to improve the taxi availability which in turn provides better business of the taxi service.**

Keywords—mapReduce, analysis, big data, performance, insights

I. Introduction

Transportation has been proved as the most dominant service in large cities. Diverse modes of transportation are accessible. In large cities in the United States and cities around the world, taxi mode of conveyance plays a foremost role and used as the best substitute for the general public use of transportation to get their necessities. For instance, by today in New York, there are nearly [1] 50,000 vehicles and 100,000 drivers are existing in NYC Taxi and Limousine Commission.

Big data technologies like map-reduce, pig, hive, apache-spark etc are becoming more and more popular. These technologies can be leveraged in the field of data mining, data warehousing, predictive analysis, etc to gain more valuable insights and increase the revenue of the organisation.

II. Implementation

The solution was provided by incorporating three different algorithmic paradigms such as map-reduce, map-reduce with combiner (local aggregation) and map-reduce with partitioner. It has been carefully ensured that the output <key, value> pair of mappers is matching the input <key, value> pair of the reducer.

A. *Map-reduce approach*

This approach contains just the map and reduce parts. Map part is responsible for creating <Text, FloatWritable> as the key-value pair. This key are made based on week days and weekends. This key contains various specifics such as overall trip count and distance for both weekdays and weekends respectively. This key is just emitted for the reducer to process

B. *Map-reduce with partioner*

This approach executes the mapper functionality in the above-mentioned pattern but varies in the output produced. The keys emitted by mappers are binned to three different buckets. This bucket contains overall averages in one bucket, hourly average of weekday and hourly average of weekend.

C. *Map-reduce with combiner(local aggregation)*

This approach has the execution in three stages, mapper, combiner and reducer. The mapper functionality is similar to the map-reduce approach mentioned above. The additional combiner is used to calculate local averages based on the similar keys emitted by the mapper. After computing, the local average, the key is emitted to reducers where the aggregation of all the keys take place.

III. Experimentation

The tests have been performed by considering various aspects like data size, performance of the algorithm, scalability factor and to support all of this, is to obtain consistent analytical insights in all the trails.

A. *Data size*

Four different sizes of 30%, 50%, 80% and 100% of the original data were considered. All the three algorithmic paradigms were executed for the data size percentage to observe the behaviour of algorithm on increasing data size.

B. *Scalability*

Four different cluster sizes of 2, 4, 6 and 8 client(slave) nodes for the experimentation to inspect the scalability bottleneck faced by the algorithm.

IV. Results

A. *Analytical Insight*

The obtained result contains an insight such as overall trips (number of records), overall distance travelled (overall total distance), average distance travelled each trip, weekday average distance travelled (Monday to Friday) and average distance travelled on weekend (Saturday and Sunday). Also, hourly average distance travelled is obtained from both weekday and weekend respectively.

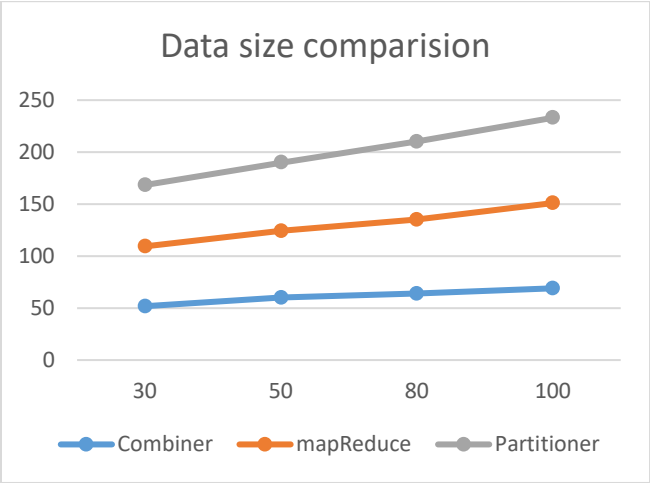
B. *Data sizes*

- Map-reduce approach [2]
- Map-reduce with combiner [2]
- Map-reduce with partitioner [2]

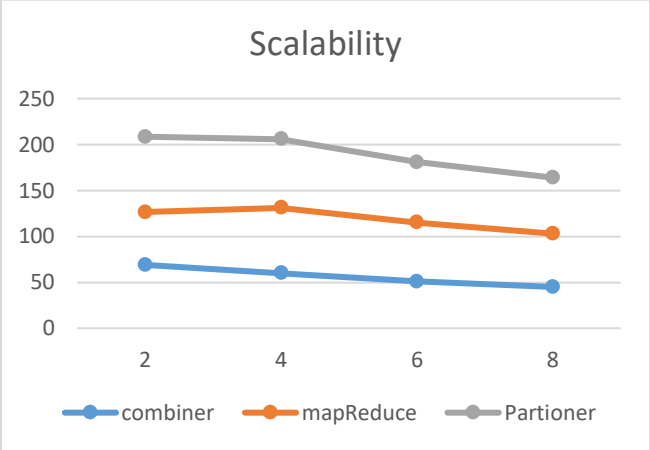
C. *Scalability*

- Map-reduce approach [3]
- Map-reduce with combiner [3]
- Map-reduce with partitioner [3]

V. Figures and Tables



[2] Table 1 – Data size comparison of three algorithm paradigm



[3] Table 2 – Scalability comparison of three algorithm paradigm

VI. Discussion

A. *Analytical insight*

The insights obtained can be used helpful in providing much better facility so that the customer loyalty remains as is. As per the analysis, people tend to travel around 3 miles during mid-day (1-8) which is the average distance travelled by anybody using the taxi service. people tend to travel longer distances in 9-12 time frame above the average distance travelled. And travel less distance during the morning.

B. *Data size*

It can be observed that map-reduce approach with combine performs the best compared to other two approaches. It can be seen that all of the three approaches, that as the size of the data set increases, the execution takes a bit longer time than the latter with smaller size of the data.

C. *Scalability*

As the size of the clusters were increased, the computational time required by the algorithm reduced. However, for much larger datasets it can be seen that as the clusters increase, speed of the execution of algorithm increase to a point and becomes constant after a certain point.

VII. Conclusion

The overall the report looks at the insights and performance of the Hadoop map reduce algorithm for New York taxi dataset. Further enhancements can thoughts in terms of increasing the data set size and considering various other features for the analysis.

References

[1] https://www1.nyc.gov/site/tlc/about/about-tlc.page