Analyzing U.S.A's flight statistics on Big Data Environments using HDFS, MySQL, Hive, HBase, Sqoop and MapReduce.

Pavan Kumar Sudhakar[1]
MSc. Data Analytics, School of Computing
National college of Ireland, Dublin.

Abstract— **In the current world, transportation is considered as one of the basic requirements in humans day to day life. Air travel has revolutionized the transportation industry and changed the mindset of human that traveling to any part of the world isnt that difficult. This massive improvement also brings in many practical difficulties to this industry, especially factors like flight delays, flight cancellations, and operations of flights by the airport. These problems are analyzed, identified and being fixed. But with the generation of a huge amount of data, it has become a struggle to store and analyze the data such data. In this project, a massive volume of flight operation data in USA has been gathered and processed in big data environments using HDFS architecture and databases like MySQL, Hive, HBase. Sqoop has been to successfully transfer the data from one database to another. MapReduce programs and Hive queries were used to analyze the flight data to address some of the problem areas in the aviation industry which could bring in more efficiency in their operation.**

Keywords: Flight, Aviation, Big data, HDFS, Hive, HBase, MySQL, MapReduce, Design Patterns

I. INTRODUCTION

In the evolution of human-kind, transportation contributes a higher percentage of significance. Out of several transportation medium, the invention of air travel has been a significant achievement. Air travel has changed the mindset of humans and made the world to look very small and easy to connect anywhere at any time. In recent years, the speed advancement of the aviation industry has made planning and operation of airplanes a tedious job, because the number of flights in operation in an airport is increasing every year. This causes confusion and sometimes leads to delay in departure of flights and sometimes to cancellation of the flights [1]. This is a major reason for the poor performance of airports which operates without proper monitoring and control. This brings in unnecessary anxiety to passengers which would get deteriorated over time. [2]

With the arrival of artificial intelligence and data science, many problem areas were identified in this domain by understanding the pattern from the data captured. Generally, any data related to the aviation industry is so big and handling such data is not that simple. While, storing such data is itself a tough process, to process such data in order to identify a pattern to solving the problems in the aviation industry is a difficult job. Hence, to apply artificial intelligence to such big raw data, storing the data in a distributed storage system is an ideal solution than keeping such a large volume of data in traditional relational databases.

But, gathering and storing such huge volume of data from different sources into a distributed storage environment has many challenges. The database selection at different stages of the storage process needs subject matter expertise and complete understanding of type of data is going to be stored. Apache Hadoop has been in the picture since a very long time to provide a one-stop solution to a distributed storage environment architecture. HDFS provides tremendous scalability and compatibility to work well other databases [3]. Databases like HBase, hive, pig were the most common databases used in a distributed storage architecture. Data transfers between these databases have been made simpler and efficient with many new tools and technologies in the market [4]. One such technology is provided by apache open source foundation itself known as Sqoop.

Ideally, these data transfer technologies require some basic ETL operations like creation of table with the same structure in destination database as in the source database, maintenance of metadata information at both databases, etc. In real-world applications, many different databases would be used together to store different information at different stages of data acquisition and processing. Usually, most of these

databases will be connected with one another to maintain synchronization and data integrity by using some kind of data transfer tools/technologies. [5] The key in any big data environment is choosing the right databases for the right data and processing the data stored inside with suitable data processing engines like Map Reduce, Spark, etc. Map Reduce is very efficient in processing stored in distributed environments and it can be coded in different programming languages like java, python, C++, etc. [6]

This research aims to identify the top canceled flights across the United States, and the routes where most of the flights were canceled, the time in air spent by the flights each month across regions and the most frequently visited airports using hive. An average delay caused by different carriers across regions and most frequently canceled flight routes using Map Reduce programs. For this research, a big data set is procured from Kaggle, this dataset includes the information related to airports, airlines, carrier information, cancellation details, diversion details, time of certain flights in air and etc, in the United States of America. By using this dataset, this research was able to answer the portrayed research questions.

II. **Literature Review:**

In [7], a big data framework has been effectively used to store the failure information of flights, and those failures were evaluated using a transient pattern matching algorithm and stored in a database in the Apache Hadoop. Hadoop mainly works on the Map-Reduce environments, and the data stored in Hadoop is distributed across its clusters. With Hadoop architecture, its easy to store both structured and unstructured data. Map reduce works by mapping the input task to a number of slave nodes to work independently on the subsets of the data. Then in the reducer stage, the outputs of all mapper jobs are combined to produce the final output.

The design of MapReduce models is in such a way that computation happens where data is stored instead of moving the data to where computation is happening [3]. Also, Hadoop is highly fault tolerant which is achieved by replicating the data and tracking the status of worker nodes polling method. This ensures continuous operation of the Hadoop environment, because the failed nodes are replaced with active nodes immediately and the jobs are re-mapped accordingly. In [3], information such as Temperature, voltage, current, surface fault class, etc, has been collected to and stored in a Hadoop environment. The data is then processed to find a transient pattern. It is then coded and labeled to apply the ANN model. The processed data is again stored in the Hadoop framework.

In [8], analysis of airplane fuel efficiency has been performed in big data environments using Hadoop and MapReduce frameworks. The visualizations were carried out by using the R package. The raw data was gathered and stored in MongoDB, which is then accessed using R to perform cleaning and store the data back into MongoDB. The cleaned data is then transferred to the HDFS environment where the data is distributed across the Hadoop clusters. The HDFS platform serves as the main storage repository in this architecture which distributes the data into further NoSQL and relational databases as per the application needs. The data from HDFS is transferred using the Sqoop tool. Processing of data happens at different databases and at different layers of the architecture as per the needs of the application.

In [5], an optimized query processing architecture for hive has been proposed. In general, whenever a query is executed in the hive, where clause is applied on entire dataset to fetch the results. This approach is bottleneck whenever the query encounters a huge table. Query partitioning techniques are employed by the hive to suppress this problem. By using such a technique, whenever a query is executed in a big table, only the data required parts of the table is executed instead of the whole table. This highly minimizes the query service time [9].
From the knowledge acquired through the previous research works, a novel architecture of a big data environment is designed to address the business queries listed in previous sections.

III. SYSTEM SPECIFICATION

Memory	4GB
Disk	80GB
CPU	4
Number of HDFS data npodes	1
Operating System	Ubuntu 18.04

TABLE I
SYSTEM SPECIFICATION

IV. METHODOLOGY

Analyzing the flight travel and delay data is an important process to carry out the airport operation smoothly. Thousands of flights fly all over the world every day and storing all flight-related information is not an easy task. The data collected/generated by this operation is very larger and it cannot be accommodated in the normal relational database for analyzing. It requires a big data platform like Hadoop, spark which is very efficient in handling such a huge volume of data. In this project the analysis of US flight information has been carried out using multiple databases like HBase, Hive and MySQL in a big data environment (Hadoop) [10] [11]. The different phases of the architecture would be covered in this section under below subsections.

- Data sourcing.
- Data pre-processing
- Data Processing and Storage.
- Visualization.

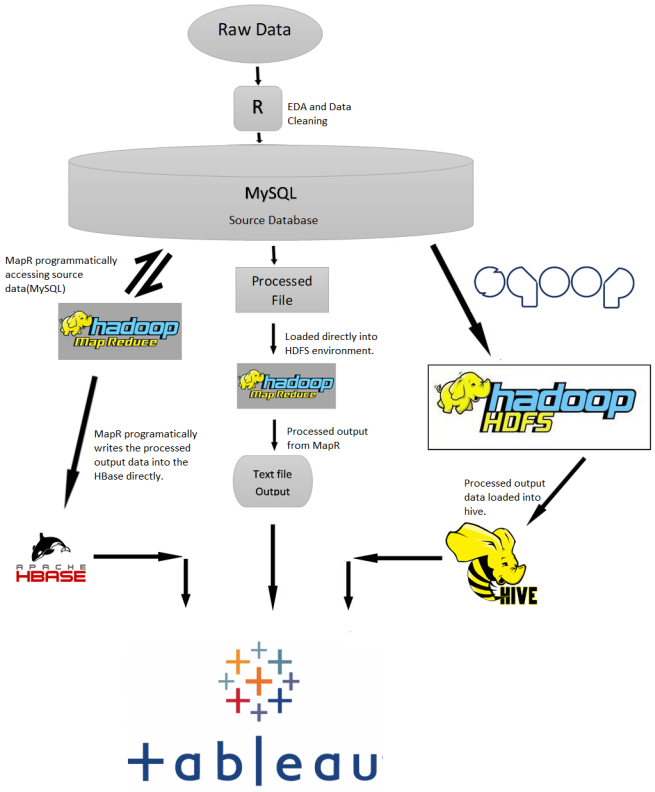

Fig. 1. **Overall Architecture of Big data environment.**

A. Data Sourcing

The dataset for this analysis is taken from Kaggle which is an opensource dataset provider. This dataset consists of year wise detailed data of all domestic airlines and airports in the United States of America, which includes the spatiotemporal data of airlines, information regarding cancellations, delays, diversions, carrier code of flights, time spent in air by each flight, etc. The data gathered was very huge (millions of rows) and it's extremely difficult to store such data in traditional databases. Overall, this dataset has 31 columns and some of them are removed as a part of pre-processing which will be discussed in the next section.

Column Name	Description
Year	Year of the flight trip
Month	Month of the flight trip
Day	Day of the flight trip
Day_of_Week	Day of the week of the flight trip.
Airline	Airline identifier
flightNum	Flight Identifier
Orgn_Airport	Starting Airport
TailNum	Aircraft Identifier
Dest_Airport	Destination Airport
ActualElapsedTime	Actual elapsed Time
CRSElapsedTime	Air_time+taxi_in+taxi_out
AirTime	Total Delay on Departure
ArrDelay	The time duration elapsed between departure from the origin airport gate and wheels off
DepDelay	Total Delay on Departure
Distance	Planned time amount needed for the flight trip
TaxiIn	The time duration elapsed between wheels-on and gate arrival at the destination airport
TaxiOut	The time duration elapsed between departure from the origin airport gate and wheels off
Cancelled	Times cancelled
DepTime	Departure time
CRSDepTime	Wheel_off - taxi_out
CRSArrTime	Wheel_off - taxi_in

TABLE II
Meta data description

B. Data Pre-Processing

In this phase, the columns which are not relevant for addressing the research question and columns that have the majority of NA/missing values were removed using R. The month column in raw data was in numeric format, this was converted into text column by replacing the number with name of the month. Special characters in the data were identified and removed as well. Some of the missing value columns were imputed using machine learning techniques (rfimpute mice). After this phase, the data has 23 columns and 1048576 rows, the metadata was listed in table II.

Tools and Technologies used.	Role in architecture.	Reason and Justification
Apache Hadoop	To store the data in a distributed environment which makes it fault tolerant and efficient in handling big data.	Highly compatible with numerous tools and available as open source.
MapReduce	Processes the data in HDFS and stores the O/P in Hbase.	Proven engine to process big data in HDFS.
Hive	Runs on top of HDFS to process big data.	Easy to construct complex queries.
Sqoop	Transferring the data from MySQL into HDFS	Works well in transferring data from one database to another.
Hbase	Used to store MapReduce outputs and serves as end database.	Easy to access data in HDFS environment.
Mysql	Source database, which gives data to hive and HDFS	Best relational database and serves well as a source database.
R	Initial data cleaning and transformation.	Easy to play with data.
Tableau	To visualize the results of business queries.	Provides extremely well-designed graphs and visualizations.
Eclipse IDE	To write java classes for MapReduce	Provides excellent user interface to write java programs.

Fig. 2. **Tools used and Justification**

C. Data Processing and Storage

In order to maintain efficiency and concurrency, three different data processing methods were designed which can process multiple queries at the same time and stores the processed outputs at different databases. Before the actual data processing, the raw data obtained from the source is cleaned and transformed in R as explained in previous sections and loaded into MySQL database. MySQL database acts as a source database for this entire architecture which can be accessed by HDFS and Hive parallelly. The data to the MySQL is loaded through a .csv file from local ubuntu machine using connection string as shown below.

mysql -u hive -p –local_infile=1 PDA -e ”LOAD DATA LOCAL INFILE ’/home/hduser/Downloads/DelayedFlights.csv’ INTO TABLE flight_input3 FIELDS TERMINATED BY ’,’”;

*1) **Method1**:* In the first method, the ability of MapReduce programs has been extensively researched and applied, where the data from MySQL is directly taken by the MapReduce programs without using any external tools. Here, HBase is used as destination database as it is extremely good at storing data as key/value pairs and can provide random read and write capabilities.

Design flow:

1) MapReduce programmatically accesses the source database (MySQL).
2) Processes the data in HDFS.
3) After processing, stores the results in byte format using ImmutableBytesWritable class in java, into HBase database.

*2) **Method2**:* An alternative approach is also designed in which the data is loaded into HDFS using connection string and MapReduce program is executed in an HDFS environment to process the data and produce the results. Then, the processed data is extracted as a text file output from the HDFS environment. This method is implemented as some of the application doesnt work well with byte format output (Method1 HBase output) and requires quick viewing of the processed data from MapR.

Design flow:

1) Data is loaded from MySQL into HDFS by using connection string.
2) Data is processed using MapReduce programs in HDFS.
3) The results are saved and extracted as a textfile for quick interpretation.

*3) **Method3**:* In this method, instead of using MapReduce programs, Hive queries are executed directly on HDFS environment to generate the outputs. Hive queries are equally powerful and much easier to construct than MapR programs. Here, the data to HDFS environment is brought by using SQOOP tool, which takes the data from source (MySQL) to HDFS efficiently and minimal data loss. Before processing, the table must be created in the hive with the same

metadata as source data to accommodate the incoming processed data.

sqoop import connect jdbc:mysql://127.0.0.1/PDA –username hive –password admin –table flight_input3 –target-dir /flight_input3 -m 1;

Design flow:

1) Data from MySQL is loaded into HDFS using SQOOP.
2) Hive queries are executed in HDFS.
3) Results are stored in Hive tables. interpretation.

D. Data Visualisation

Data processed in big data environments are extremely complex and sometimes has a wide range of values, this could be very difficult for end users to interpret. An appropriate visualization is very essential to present the results to the business users to understand what the underlying data is. In this research, Tableau is used to display all the processed outputs of research questions. The outputs are taken as textfile and loaded into a tableau for better visualization.

V. RESULTS

BI Query1: Top cancelled flights routes across united states. This query was addressed using MapR program, in where a Numerical summarization Pattern was used to group the canceled flights based on the flight number and sorted in descending order to obtain the top canceled flights. A customized record reader function was developed and used to fetch the data from MySQL. The format of the data is also validated and transformed by using MysqlinputFormat. The 3 represents the list of flight routes which had maximum cancellations occurred in USA domestic airports.

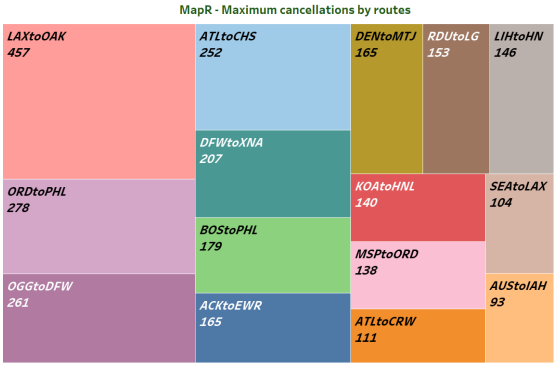

Fig. 3. **Top cancelled flight routes.**

BI Query2: This query is analyzed using MapR program. Here, the average delay caused by all flight carriers (Airways) was identified and listed in order using Numerical summarization pattern. This helps in understanding the delay caused and helps the airways to improve their performances in the future. Figure 4 shows the average delay time conceded by different carriers where Hawaiian airways stand out with negative average delay.

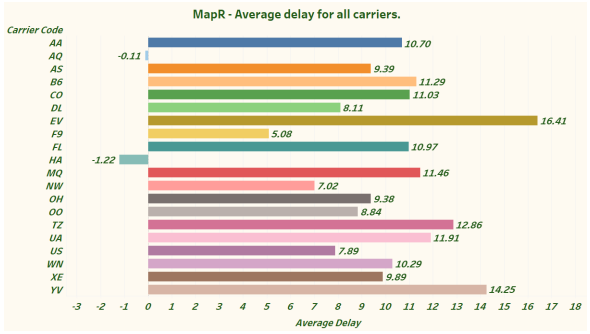

Fig. 4. **Average delay time conceded by different carriers.**

BI Query3: Highest cancelled flights by month. This query is addressed using a hive query. Here the canceled flag is used to identify whether a flight is canceled or not and those which got canceled are counted and grouped by month to identify the top canceled flight by each month. Figure 5 depicts the flights that have been canceled the most every month. Flight F511 has been canceled most of the times throughout first 2 quarters.

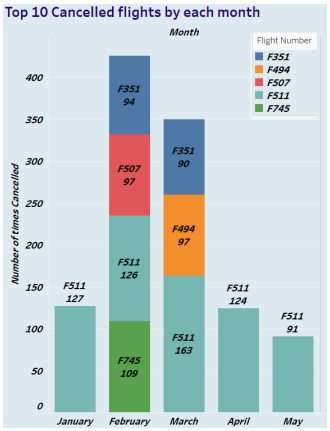

Fig. 5. **Most cancelled flights by month.**

BI Query4: Top operating flights (i.e in maximum time in air) For this query, the data is processed in

hive, where the sum of flights time spent in the air was grouped with respective flight number and by month, then sorted in descending order to get the top operating flights. From figure 6 it can be observed that F15 has been a constant flyer and remained in the air for most of the time.

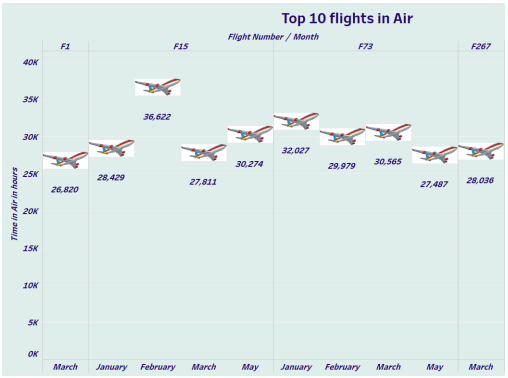

Fig. 6. **Top performing flights.**

BI Query5: Most visited airport destinations This query helps to identify the most effective operating airports in the USA and anticipate the number inflow and outflow flights in the future so that necessary arrangements shall be made. Figure 7 displays the most frequently visited airports in the USA by color intensity, it can be seen that airports in Texas have been visited most number of times by all domestic flights.

Fig. 7. **Most visited Airport destinations.**

BI Query6: Leading flight carriers in operation in United States domestic operations

This query helps in identifying the leading airways (flight carriers) in operation in the USA, which can be used to give special privileges as it is generating more revenue to the airport than any other airways. This query is addressed using hive platform where the number of uncancelled flights was chosen and counted which is then grouped based on the carrier code. Figure 8 shows the leading flight carrier in the USA in terms of operating more number of flights. The SouthWest Airlines has an outstanding record for the first 2 quarters by operating over 2 million flights.

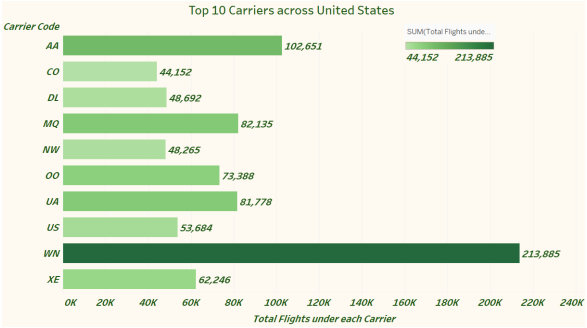

Fig. 8. **Carrier performances in terms of flight operations.**

VI. CONCLUSION & FUTURE WORKS

The overall research with flight data in a big data environment using different non-relational and relational databases have helped in identifying various underlying patterns in the data acquired, which is then used to address multiple business queries. It is observed from the query results that flights between the route Los Angeles and Oakland international airport were the most canceled. Hawaiian airlines are the best operators in the United States with negative delay in flight operation while Atlantic Southeast Airlines being the worst with an average delay of 16.14 hours in its flight operations. Flight F511 has been the most canceled flight in almost all months than any other flights in the USA. Flight F15 has been a top performer with maximum air time in the months of January, February, March, and May. Texas and California have been the most visited airport destinations in USA domestic operations. The Southwest airlines have been the most operated flight carrier with 213,885 flights operated between January to May. The processing of the above queries may look simple but the amount of data that is processed to formulate the results are very high. With the help of distributed data storage and processing architecture like Hadoop and no-SQL databases, these large volumes of data can be processed without much effort or system requirements. Though MapReduce can process such huge volumes of data, it still lags the operation speed which can be achieved by other technologies like a

spark, which processes the data in memory. But spark requires high-end system configurations. It is a trade-off between processing speed and system requirements between Spark and MapReduce engines in processing such big data.

REFERENCES

[1] B. Cox, W. Jemiolo, and C. Mutel, "Life cycle assessment of air transportation and the swiss commercial air transport fleet," *Transportation Research Part D: Transport and Environment*, vol. 58, pp. 1–13, 2018.

[2] B. Yu, Z. Guo, S. Asian, H. Wang, and G. Chen, "Flight delay prediction for commercial air transport: A deep learning approach," *Transportation Research Part E: Logistics and Transportation Review*, vol. 125, pp. 203–221, 2019.

[3] M. C. Srivas, P. Ravindra, U. V. Saradhi, A. A. Pande, C. G. K. B. Sanapala, L. V. Renu, V. Vellanki, S. Kavacheri, and A. A. Hadke, "Map-reduce ready distributed file system," Dec. 27 2018. US Patent App. 16/116,796.

[4] R. Bharti and D. Gupta, "Recommending top n movies using content-based filtering and collaborative filtering with hadoop and hive framework," in *Recent Developments in Machine Learning and Data Analytics*, pp. 109–118, Springer, 2019.

[5] D. Vohra, "Using apache sqoop," in *Pro Docker*, pp. 151–183, Springer, 2016.

[6] K. Neshatpour, M. Malik, A. Sasan, S. Rafatirad, and H. Homayoun, "Hardware accelerated mappers for hadoop mapreduce streaming," *IEEE Transactions on Multi-Scale Computing Systems*, vol. 4, no. 4, pp. 734–748, 2018.

[7] A. Banu, R. Sangeetha, and M. Nanda, "Big data application for analysis of flight test data," in *2017 First International Conference on Recent Advances in Aerospace Engineering (ICRAAE)*, pp. 1–6, IEEE, 2017.

[8] M. Li and Q. Zhou, "Industrial big data visualization: A case study using flight data recordings to discover the factors affecting the airplane fuel efficiency," in *2017 IEEE Trustcom/BigDataSE/ICESS*, pp. 853–858, IEEE, 2017.

[9] A. Elsayed, M. Shaheen, and O. Badawy, "Caching techniques for flight delays prediction in big data using sparkr," 2018.

[10] Y. Xie, A. M. Q. Farhan, and M. Zhou, "Performance analysis of hadoop distributed file system writing file process," in *2018 International Conference on Intelligent Autonomous Systems (ICoIAS)*, pp. 116–120, IEEE, 2018.

[11] N. Das, S. Paul, B. B. Sarkar, and S. Chakrabarti, "Nosql overview and performance testing of hbase over multiple nodes with mysql," in *Emerging Technologies in Data Mining and Information Security*, pp. 269–279, Springer, 2019.