Flight Data Analysis Using PIG

Ishan Meena, R. Aravind, Vijayditya Sarker, Nandhini

***Abstract*: Precise prediction of passenger flow is very important for any company to create their business policies. The passenger analysis uses key technologies that is transmission of data dynamically, huge amount of data storage, fusing of data through multiple sources, data-mining and other analysis. With the use of visualisation, data prediction and decision making, the complete set of data (authorities, passengers) can create their own goals and perspectives. Therefore, the research provides both, accurate information about the transport services to common citizens and at the same time specify business models for lower tier and higher tier companies alike.**

***Index Terms*: PIG, Hadoop, Big Data, SQL**

I. INTRODUCTION

Urban traffic includes a variety of elements such as cars, trucks, buses, taxis, public administration, transport interchange, Infrastructure of the traffic and air travel. The Air Traffic in recent years have drastically increased and hence requires more assistance and directory. As there is increase in the number of passengers, there is a need for a better data storage and data analysis. With the increase in the number of passengers, the time taken for travel is also taken into considerable account. As the passenger doesn't want to waste any time by booking a flight, it is evident that he wants to avoid the waiting time at the airports too. To add to this, the waiting time in queue also eats up a significant amount of time. Due to this, not only the passengers but also the airport agencies are affected by this. Therefore, many agencies and companies have done analysis and created methods to overcome these problems and help the customer. As the data is increasing drastically day by day, the need for a better system to handle and analyze the data is required. The existing systems collect data from helicopters, satellites, planes etc. and the analysis is done using traditional means such as SQL. The collected data is used for data mining and visualization purpose. The data collected has a lot of unwanted data which increases complexity. These methods are focused more on the cloud storage & transportation, rather than on mining and other applications. The performance and efficiency of the system is compromised due to the large data sets. The point of view of the paper is purely from business perspective. A huge volume of Big Data is in crude form that is structureless information, large volume of a single type of data, administration in flight analysis are information obtaining, large amount of data, managing the data, testing the data, and data representation. Notwithstanding the traits of enormous information specified above, it is fundamental that instruments exist for representation and comprehension of the data and relations between the information present in the datasets, which is called, Business Insight (BI). This requires information stockpiling and administration, equipment and programming assets, proper space learning, and new techniques and advances. Joining enormous information with examination can give a significant preferred standpoint to settle on auspicious and efficient choices identified with

A) Cost
B) Time
C) Item Improvement
D) Enhancement

A great amount of data is captured and used in different configurations (organized, semi-organized and unstructured), from several sources (sensors, machines, applications, web, IoT) and checked by the associations. The information is captured, kept aside or is taken care by constant or clusters with the help of mechanical procedures or calculations. Implementation of these plans change for different areas along with time, from avionics to automobile industries, taking care of all the account and capital conjectures, correlation, utilities and mining, government, hospitality industry, protection, retail, innovation, and so on. It is important for these professions to make most out of these little signs from a few important data sources both organized and unstructured. It conveys an ongoing effect for a simple, efficient and powerful basic leadership. The paper deals with Commercial Aviation data for the improvements of service, and makes use of Latin Pig Scripts which are far more efficient than normal SQL Queries, and can work on unstructured as well as structured data. The data from the previous years is visualized and according to the trends the future plans and services are made according to the same. The proposed system is available for both business as well as individual users. Data Mining & Prediction through Visualization is taken care of more than the storage and transportation which makes it more efficient. The proposed system is far more efficient than the traditional methods of data mining and processing. Structured as well as Unstructured data is taken in to account.

II. LITERATURE SURVEY

- Prediction for Air Route Passenger Flow Based on a Grey Prediction Model-

The method used by this paper is regression analysis and grey prediction. It predicts the passenger flow of an air route which guides the airline company to estimate the passenger flow and thus helping in making better sales policy.

Manuscript published on 30 April 2019.
***** Correspondence Author (s)

Ishan Meena, Computer Science and Engineering, SRM Institute of Science & Technology, Chennai, India.
R. Aravind, Computer Science and Engineering, SRM Institute of Science & Technology, Chennai, India.
Vijayditya Sarker, Computer Science and Engineering, SRM Institute of Science & Technology, Chennai, India.
Ms. Nandhini, Computer Science and Engineering, SRM Institute of Science & Technology, Chennai, India.

Its major drawback is it doesn't work efficiently with big data sets.

- A Kind of Novel ITS Based-on Space-Air-Ground Big-Data-

The method used by this paper is dynamic data transmission, multi-source data fusion. It provides accurate transportation information services for the citizens. The major drawback is that It uses complex map reduction algorithms.

- Application of Big Data Visualization in Passenger Flow Analysis of Shanghai Metro Network-

The method used here is Cluster analysis. It provides new means for passenger flow analysis and operation aid decision making (ADM). The major drawback is it doesn't work efficiently with big data sets.

III. INTRODUCTION OF BIG DATA VISUALIZATION

A significant feature of Big Data Era is the rise of data visualization. Its basic idea is to show attribute values with multidimensional graphs, i.e., analyze data by using graphical methods from different dimensions. Big data visualization is used to create a shortcut to perceive dataset quickly. It is gradually becoming a convincing way to show and communicate complex data. Low visualization tools are used such as Excel, Crystal Report, Fine Report. Big data visualization often uses high visualization framework like Mat Lab, Map box, Tableau, Echart.js, Highchart.js, Antv.js, D3.js. In the field of transportation (road traffic, metro passenger flow, traffic accident and mobile decision, air traffic), studies on Air traffic data visualization have been greatly expanded on the depth and breadth. The analytics of traffic data visualization are from traffic flow to traffic incident, from pure traffic data of air reflection to multi-source data of rich social semantics, from traditional PC visualization to new visual display medium. Big data visualization can be used to find out correlations among various factors form the massive traffic data to realize its application value. The high visualization framework Echart.js and D3.js are mainly used in this paper to study the applications of big data visualization in flight data analysis.

IV. BRIEF INTRODUCTION OF APACHE PIG

Apache pig is a very high-level platform for designing and creating programs that run on Hadoop framework. The language used in Apache PIG is PIG Latin. With the help of MapReduce technique Apache PIG completes its Hadoop related works. Pig Latin has a function called UDFs (User defined functions) which lets its user write program in any language like python, JavaScript, Ruby etc, and then it can directly call from the language. PIG Latin allow its user to include user code at any point of time. Which is helpful in pipeline development. As PIG Latin is procedural so it naturally fits into the pipeline paradigm.

There are three ways to execute PIG's script-

1. Grunt shell

It is an interactive part which helps to execute all PIG scripts.

2. Script File

Here, we write all PIG commands in a script file and it is executed by the PIG server.

3. Embedded script

This part allows user to use any language to program, which can be later embedded in PIG Latin script file. Then, later the Script file can be executed.

V. BRIEF INTRODUCTION OF HADOOP FRAMEWORK

Hadoop Framework is a collection of open source software tools that use the connectivity (network) of many computers to handle the data and analysis of the data. The Hadoop Distributed File System (HDFS) is an important and most essential data collecting framework used by Hadoop applications. It contains NameNode/The Master and DataNodes/The Slave model to execute the framework called Hadoop Distributed File System. It helps to get data over highly adaptable Hadoop clusters in a very efficient manner. Hadoop process has 5 daemon processes namely-

NameNode - In here, Metadata (data about the data) is stored for HDFS.RAM or Hard-Disks can be used to store the Metadata. A cluster will be generally be compromising of a single NameNode. If the NameNode crashes then and only then Hadoop framework the system will fail.

Secondary NameNode - It acts as a backup for NameNode. The data present in NameNode and the secondary NameNode is practically the same. If the name node somehow fails then it's use comes into picture.

DataNode - Data and actual user files are stored on DataNode. According to one's data size DataNode can be increased or decreased in definite interval of times DataNode communicates with NameNode.

Job Tracker - NameNode and DataNodes save the points of interest and the proper information on the File System. The information is used to process according to the users' requirements. A code is written to process the information. The processing of data is done by using MapReduction. The MapReduce Engine sends the code to DataNodes, which makes the jobs in different nodes run alongside each other. These are to be kept track of by the JobTracker.

Task Tracker - The tasks undertaken by Job Trackers are properly performed by Task Trackers. One task tracker is assigned to a DataNode. Task Tracker requests the Job Trackers about the status of the undertaken job.

Master and slaves receive a quick bolster of information from HDFS as it is combined with MapReduce, an automatic system which helps to handle information and for accessing data at a higher rate. With the help of parallel processing the system is made very effective, data is separated into partitioned squares and is stored in various nodes. Hadoop Distributed File System is highly efficient and fault tolerant. On various occasions each bit of information is duplicated and reproduced by the framework, it also conveys duplicates in the singular hubs, it also puts one duplicate in an alternate rack.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication (BEIESP)

Inside the group we can find the information that crashed in the hub. Master and Slave architecture is followed by HDFS. A HDFS group consists of a NameNode, the master server manages access to the data and deals with the data stored by the help of authorized users in the Hadoop environment. Google File System (GFS) inspired the core of Hadoop File Distributed System (HDFS). In Google technical papers a restrictive document was laid out. General Parallel File System lifts i/o by writing blocks of data into disks in parallel which provide efficiency. HDFS echoes POSIX configuration style, though it is not Portable Operating System.

VI. SYSTEM ARCHITECTURE

Apache pig is used to simplify the processing of huge datasets. Writing a proper code requires a lot of technical skills and vast knowledge of the programming language. This process is highly time taking and inefficient. Pig technique provides better and complex nested data structure that makes it different from the regular SOL queries. It helps in decreasing the length of the code by using multiple query concept. Pig can be used to replace 200 lines of java code with mere 10 lines of pig code. Pig consists of two parts, first is the Pig Latin Language and the second is the Pig environment to the executable Pig codes.

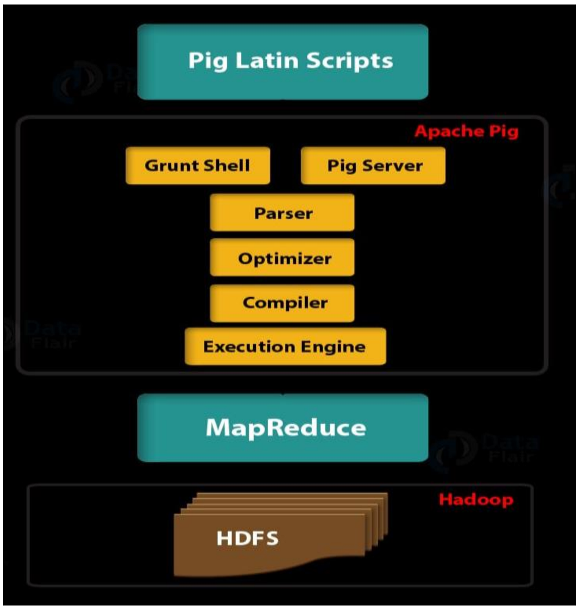


Figure 1: Pig Architecture

The major components of Apache Pig can be given as -

- **Parser -** The initial stage of Apache Pig is Parser. All the Pig scripts are first handled by the parser. The input script is checked for syntax and correct data types and other details. The output of the parser is directed acyclic graph (DAG). In the DAG, the logical operators of the script are labelled as nodes and the data flows as edges.

- **Optimizer -** The logical plan that is the DAG is produced for every single line and then the semantic check is done for the script. It uses the basic parsing. It is later sent back to the logical analyzer and optimizer and then it performs the optimizations like pushdown and projections.

- **Compiler -** The job of the compiler is to translate the whole script. The optimized logical plan is then changed to a more understandable MapReduce job.

- **Execution engine -** the execution engine makes sure that the MapReduce jobs to Hadoop are all in a sequence and assorted manner. These then help to produce the desired results. They are usually implemented on a HDFS.

VII. ALGORITHM USED IN THIS PAPER

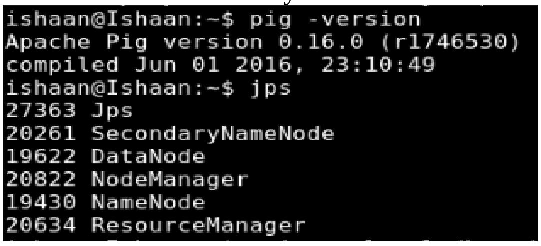
Map Reduction algorithm has two major tasks, Map and Reduce

- Mapping - Obtained by Mapper Class
- Reduction - Obtained by Reducer Class

MapReduce uses different calculations to separate an errand into small parts and roll them into different frameworks. MapReduce Calculations help in sending the Map and Reduce errands to appropriate servers in a large bunch. The tasks are implemented and executed side by side in all the different nodes and the result is returned to the user. Hadoop uses MapReduce algorithm to create tasks, which are called jobs which in turn can be executed without depending on any other factor, on different cluster while the result is fetched back to a single node as output.

VIII. IMPLEMENTATION

The Implementation of the proposed model has been done on a fully functional Ubuntu system with Latest Apache Hadoop, Apache Pig and An Apache Web Server for remote access via Shell In A Box Utility.

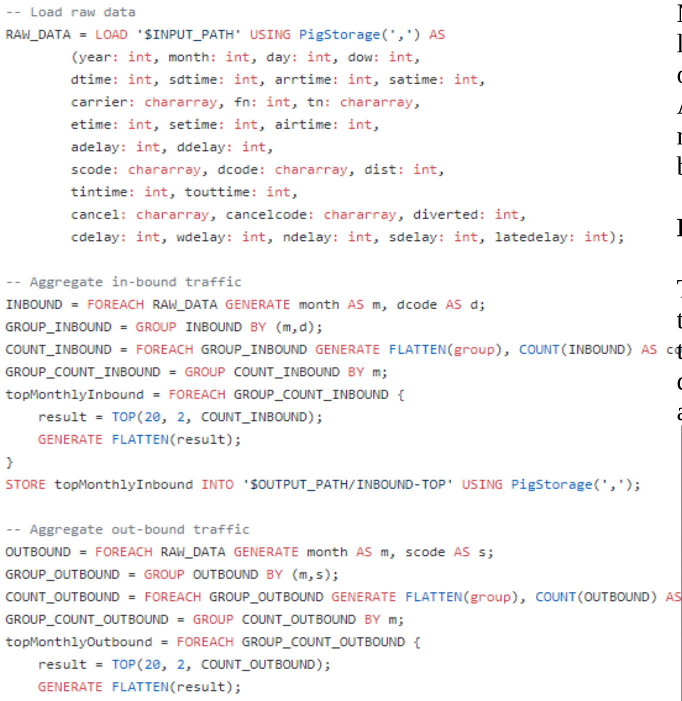


```
ishaan@Ishaan:~$ pig -version
Apache Pig version 0.16.0 (r1746530)
compiled Jun 01 2016, 23:10:49
ishaan@Ishaan:~$ jps
27363 Jps
20261 SecondaryNameNode
19622 DataNode
20822 NodeManager
19430 NameNode
20634 ResourceManager
```

Figure 2: Hadoop Daemons

The database being used for the simulation of the working of the Pig Latin Script to mine data out of the dataset is taken from The U.S. Department of Transportation's (DOT) Bureau of Transportation Statistics (BTS) which records the different statistics of Airline Data such as the number of on-time, delayed, cancelled and diverted flights.

Analysing these datasets, and using them for the purpose of data mining which give us insight via trends can help us predict various things such as the busiest time of the year when one is likely to face flight delays. Thus, using the above one can provide insight which can help both the consumers and the flight companies. Below is an example Pig Script which finds the Top 20 Busiest Airports in The United States from the given dataset.



```
-- Load raw data
RAW_DATA = LOAD '$INPUT_PATH' USING PigStorage(',') AS
        (year: int, month: int, day: int, dow: int,
        dtime: int, sdtime: int, arrtime: int, satime: int,
        carrier: chararray, fn: int, tn: chararray,
        etime: int, setime: int, airtime: int,
        adelay: int, ddelay: int,
        scode: chararray, dcode: chararray, dist: int,
        tintime: int, touttime: int,
        cancel: chararray, cancelcode: chararray, diverted: int,
        cdelay: int, wdelay: int, ndelay: int, sdelay: int, latedelay: int);

-- Aggregate in-bound traffic
INBOUND = FOREACH RAW_DATA GENERATE month AS m, dcode AS d;
GROUP_INBOUND = GROUP INBOUND BY (m,d);
COUNT_INBOUND = FOREACH GROUP_INBOUND GENERATE FLATTEN(group), COUNT(INBOUND) AS count;
GROUP_COUNT_INBOUND = GROUP COUNT_INBOUND BY m;
topMonthlyInbound = FOREACH GROUP_COUNT_INBOUND {
    result = TOP(20, 2, COUNT_INBOUND);
    GENERATE FLATTEN(result);
}
STORE topMonthlyInbound INTO '$OUTPUT_PATH/INBOUND-TOP' USING PigStorage(',');

-- Aggregate out-bound traffic
OUTBOUND = FOREACH RAW_DATA GENERATE month AS m, scode AS s;
GROUP_OUTBOUND = GROUP OUTBOUND BY (m,s);
COUNT_OUTBOUND = FOREACH GROUP_OUTBOUND GENERATE FLATTEN(group), COUNT(OUTBOUND) AS
GROUP_COUNT_OUTBOUND = GROUP COUNT_OUTBOUND BY m;
topMonthlyOutbound = FOREACH GROUP_COUNT_OUTBOUND {
    result = TOP(20, 2, COUNT_OUTBOUND);
    GENERATE FLATTEN(result);
```

Figure 3: Pig Script

Working

Assigning Jobs to DataNodes

The program takes inputs from the comma separated values dataset file and inputs all the data into the Hadoop distributed file system which is then passed on to different DataNodes to be processed. Every DataNode perform operations on particular part of the data set called a “Job”. The JobTracker tracks all of these operations and ensures proper operations and integrity.

Map Reduction Phase

The HDFS performs map reduction algorithm onto the file in various DataNode which on completion send back the results back to the NameNode which compiles back the incoming data from different sources to a single file which can be later viewed by the user. Generally, MapReduce paradigm is based on sending the computer to where the data resides!

MapReduce program executes in three stages, namely map stage, shuffle stage, and reduce stage.

- **Map stage** − The map or mapper’s job is to process the input data. Generally, the input data is in the form of file or directory and is stored in the Hadoop file system (HDFS). The input file is passed to the mapper function line by line. The mapper processes the data and creates several small chunks of data.
- **Reduce stage** − This stage is the combination of the Shuffle stage and the Reduce stage. The Reducer’s job is to process the data that comes from the mapper. After processing, it produces a new set of output, which will be stored in the HDFS.

While performing a MapReduce job, Hadoop sends the Map and Reduce tasks to the servers where it seems the data fit in the cluster.

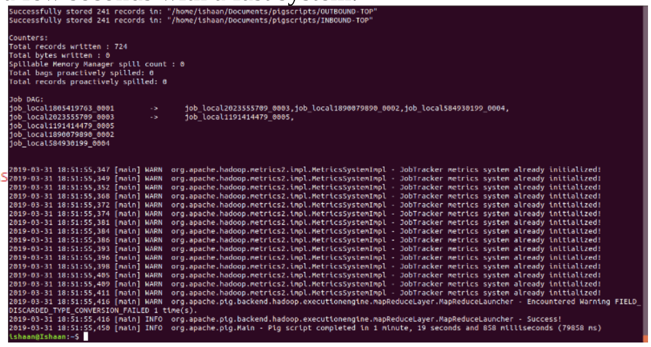
The framework manages all the details of data-passing such as issuing tasks, verifying task completion, and copying data around the cluster and between the nodes.

Most of the computing takes place on nodes with data on local disks that reduces the network traffic and increasing overall performance.

After completion of the given tasks, the cluster collects and reduces the data to form an appropriate result, and sends it back to the Hadoop server.

Latin Script Working [Busiest airports]

The Latin Script first calculates the in-bound traffic on all of the airports and then calculates the outbound traffic on all of the airports and combine them for matching airports. This is done for around 80 lakh records which are processed in under a few seconds with a fast system.



```
Successfully stored 241 records in: "/home/ishaan/Documents/pigscripts/OUTBOUND-TOP"
Successfully stored 241 records in: "/home/ishaan/Documents/pigscripts/INBOUND-TOP"

Counters:
Total records written : 724
Total bytes written : 0
Spillable Memory Manager spill count : 0
Total bags proactively spilled: 0
Total records proactively spilled: 0

Job DAG:
job_local1805419763_0001        ->      job_local2023555709_0003,job_local1890079890_0002,job_local584930199_0004,
job_local2023555709_0003        ->      job_local1191414479_0005,
job_local1191414479_0005
job_local1890079890_0002
job_local584930199_0004


2019-03-31 18:51:55,347 [main] WARN  org.apache.hadoop.metrics2.impl.MetricsSystemImpl - JobTracker metrics system already initialized!
2019-03-31 18:51:55,349 [main] WARN  org.apache.hadoop.metrics2.impl.MetricsSystemImpl - JobTracker metrics system already initialized!
2019-03-31 18:51:55,352 [main] WARN  org.apache.hadoop.metrics2.impl.MetricsSystemImpl - JobTracker metrics system already initialized!
2019-03-31 18:51:55,368 [main] WARN  org.apache.hadoop.metrics2.impl.MetricsSystemImpl - JobTracker metrics system already initialized!
2019-03-31 18:51:55,372 [main] WARN  org.apache.hadoop.metrics2.impl.MetricsSystemImpl - JobTracker metrics system already initialized!
2019-03-31 18:51:55,374 [main] WARN  org.apache.hadoop.metrics2.impl.MetricsSystemImpl - JobTracker metrics system already initialized!
2019-03-31 18:51:55,381 [main] WARN  org.apache.hadoop.metrics2.impl.MetricsSystemImpl - JobTracker metrics system already initialized!
2019-03-31 18:51:55,384 [main] WARN  org.apache.hadoop.metrics2.impl.MetricsSystemImpl - JobTracker metrics system already initialized!
2019-03-31 18:51:55,386 [main] WARN  org.apache.hadoop.metrics2.impl.MetricsSystemImpl - JobTracker metrics system already initialized!
2019-03-31 18:51:55,393 [main] WARN  org.apache.hadoop.metrics2.impl.MetricsSystemImpl - JobTracker metrics system already initialized!
2019-03-31 18:51:55,396 [main] WARN  org.apache.hadoop.metrics2.impl.MetricsSystemImpl - JobTracker metrics system already initialized!
2019-03-31 18:51:55,398 [main] WARN  org.apache.hadoop.metrics2.impl.MetricsSystemImpl - JobTracker metrics system already initialized!
2019-03-31 18:51:55,405 [main] WARN  org.apache.hadoop.metrics2.impl.MetricsSystemImpl - JobTracker metrics system already initialized!
2019-03-31 18:51:55,409 [main] WARN  org.apache.hadoop.metrics2.impl.MetricsSystemImpl - JobTracker metrics system already initialized!
2019-03-31 18:51:55,411 [main] WARN  org.apache.hadoop.metrics2.impl.MetricsSystemImpl - JobTracker metrics system already initialized!
2019-03-31 18:51:55,416 [main] WARN  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - Encountered Warning FIELD_
DISCARDED_TYPE_CONVERSION_FAILED 1 time(s).
2019-03-31 18:51:55,416 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - Success!
2019-03-31 18:51:55,450 [main] INFO  org.apache.pig.Main - Pig script completed in 1 minute, 19 seconds and 858 milliseconds (79858 ms)
ishaan@Ishaan:~$
```

Figure 4: Pig Latin Script Being Executed

The generated list is then processed to find out the top 20 busiest airports and the time of month when the airports were busy which can later be visualized to get better insight. Finding the same for multiple years in succession helps one to find the busiest airports in various months which can be used to change company policies for better profits. The same can be used by the consumers to find out best time to travel around the year.

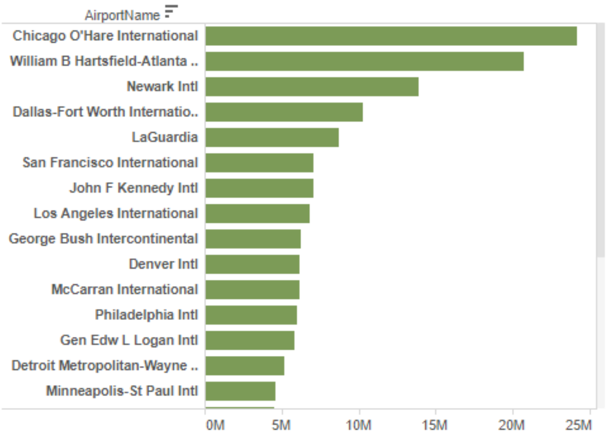


Figure 5: Busiest Airports

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication (BEIESP)
© Copyright: All rights reserved.

Similarly, you can find out the busiest routes, most delayed flights, most common reason for the delayed flights etc.
Below is a visualization of the cancellation trend.
From the visualized data you can clearly see that the greatest number of cancelled flights are in February, followed by December, this one can conclude that it not the best time to travel. Similarly, the least number of flights cancelled are in November thus making it good time to travel.

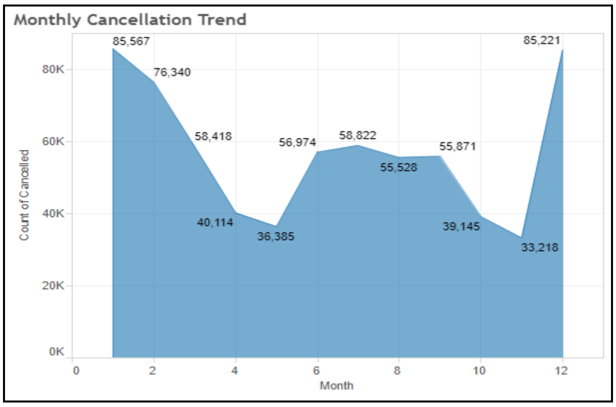


Figure 6: Number of Flights Cancelled (Monthly)

Below is an example of weekly analysis of the flight delayed which can be used by customers to book their flights at the day with least possible delays.

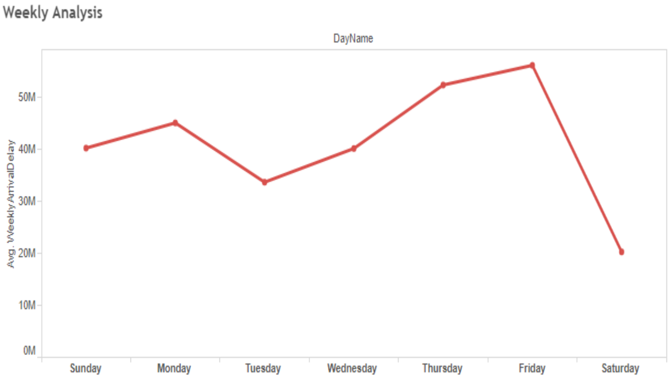
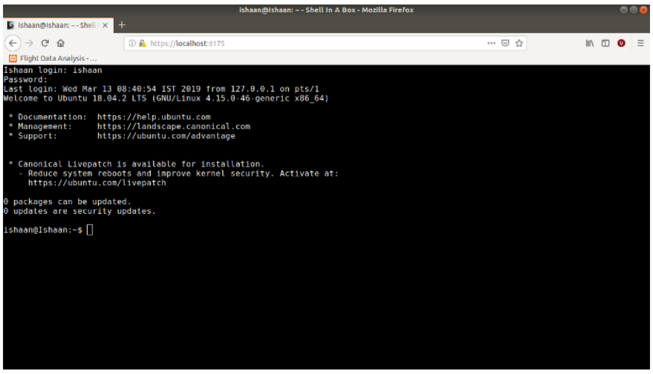


Figure 7: Number of Flights Cancelled (Daily)

An Individual can use the generated data to find out the best month and day to fly to a specific location for himself or herself. Not only that, one can even find out which Airport will be the best for him to fly from and which Airline should he or she choose to avoid any problems in that particular month.The data can also be provided by various airlines to promote their company and improve user experience. It can also be used to reform existing policies and fares of airlines to their advantage.
The above implementation is remotely accessible using ShellInABox utility where a user can remotely perform operation on the dataset.
The remote user can enter the credentials and connect to the remote system and perform various tasks such as executing a pig script and checking its output, which can further we used for visualization.


ishaan@Ishaan: ~ - Shell In A Box - Mozilla Firefox
ishaan@Ishaan: ~ - Shell
https://localhost:6175
Flight Data Analysis - ...

```
Ishaan login: ishaan
Password:
Last login: Wed Mar 13 08:40:54 IST 2019 from 127.0.0.1 on pts/1
Welcome to Ubuntu 18.04.2 LTS (GNU/Linux 4.15.0-46-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage


 * Canonical Livepatch is available for installation.
   - Reduce system reboots and improve kernel security. Activate at:
     https://ubuntu.com/livepatch

0 packages can be updated.
0 updates are security updates.

ishaan@Ishaan:~$ 
```

Figure 8: Remote access portal

IX. FUTURE WORK

The future work includes the combining of more data sources. These data sources can be the datasets provided by the companies. It also includes comprehensive benchmarking of all information to guarantee their versatility in a creation in aviation industry. This paper uses Apache Pig technique that proves to be a very efficient and an easy way to manage data flow by analyzing and surveying the flight details along with passenger details. In future we can use more complex data congregating techniques and map reduce standards for examining different flight details from different aviation industries.

X. CONCLUSION

In the era of Big Data, Data Visualization is one of the most important applications. Using data visualization, data mining is greatly improved. Data can be more effectively and intuitively can be characterized and visualized using Big Data. In flight data analysis using PIG, has achieved great results by analyzing, predicting, and visualizing data. It helps companies to improve their sales and passenger flow using `more efficient methods like Latin Pig Scripts than normal SQL Queries. It uses both structured and unstructured data. The data from previous years is visualized and according to the trends the future plans and services are made according to the same. Through visualization is taken care of more than the storage and transportation which makes it more efficient.

REFERENCES

1. Huang Zhiyuan, Zhang Liang, Xu Ruihua,; Application of Big Data Visualization in Passenger Flow Analysis of Shanghai Metro Network ; 2017 2nd IEEE International Conference on Intelligent Transportation Engineering.
2. Hari Bhaskar Sankaranarayanan, Gaurav Agarwal; An Exploratory data analysis of airport wait times using big data visualisation techniues; 2016 International Conference on Computational Systems and Information Systems for Sustainable Solutions.
3. Z. Fengxia, Z. Yaozhong, and L. Yingying, "Research on Hybrid Layered Architecture of Command and Control System for Space-air-ground Collaboration," *J. Acad. Equip.*, vol. 24, no. 5, pp. 74–77, Oct. 28, 2013.
4. H. S. Zhaohui, et al., "Vehicle change detection from aerial imagery using detection response maps," in *Proc. SPIE, Int. Society Optical Engineering, Geospatial InfoFusion Video Analytics IV Motion Imagery ISR Situational Awareness II*, 2014, vol. 9089.
   S. Jiwon and T. Walter, "Future dual-frequency GPS navigation system for intelligent air transportation under strong ionosphere scintillation,"*IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 5, pp. 2224–2236, Oct. 1, 2014.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication (BEIESP)

5. A. Arghavan, V. R. Monir, J. M. De La Garza, and R. M. Buehrer, “Improving GPS-based vehicle positioning for intelligent transportation systems,” in Proc. IEEE Intelligent Vehicles Symp., 2014, pp. 1023–1029.
L. Han and K. Wu, “Multifunctional Transceiver for Future Intelligent Transportation Systems,” IEEE Trans. Microwave Theory Tech., vol.
6. 59, no. 7. pp. 1879–1892, 2011.
7. P. Anuj, V. Kimon, and K. Michael, “Generating traffic statistical profiles
8. using unmanned helicopter-based video data,” in Proc. IEEE Int. Conf.
9. Robotics Automation, 2007, pp. 870–876.
10. W. P. Hyoung, Y. Y. Il, and J. R. Lee, and J. Haengjin, “Study on big data center traffic management based on the separation of large-scale data stream,” in Proc. Seventh Int. Conf. Innovative Mobile Internet Services Ubiquitous Computing, 2013, pp. 591–594.
11. Q. Q. Li, B. Lei, Y. Yu, and R. Hou, “Real-time highway traffic information extraction based on airborne video,” in Proc. 12th Int. IEEE Conf. Intelligent Transportation Systems, St. Louis, MO, 2009, pp. 1–6.
12. Z. Li, C. Chen, and K. Wang, “Cloud computing for agent-based urban transportation systems,” IEEE Intell. Syst., vol. 26, no. 1, pp. 73–79, 2011.
13. J. Yu, F. Jiang, and T. Zhu, “RTIC-C: A big data system for massive traffic information mining,” in Proc. Int. Conf. Cloud Computing Big Data,2013, pp. 395–402.
14. S. Bitam and A. Mellouk, “ITS-cloud: Cloud computing for intelligent transportation system,” in Proc. IEEE Global Communications Conf., 2012, pp. 2054–2059.