The Role of Predictive Big Data Analysis of Airline Data Report by using Hive

Ankaiah.G\textsuperscript{1}, Rajeshkumar.P\textsuperscript{2}, Mun hemisphere\textsuperscript{3}

\textsuperscript{1}Student, Master of Computer Applications, SKIIMS, Srikalahasti, Andhra Pradesh India
\textsuperscript{2}Research Scholar, Computer science, Bharathiar University, Coimbatore, India
\textsuperscript{3}Asst. Professor Master of Computer Applications, SKIIMS, Srikalahasti, Andhra Pradesh India

Abstract: - The analysis of the airline data set is performed using Cloudera which runs Hadoop in the cloud. Hive and Hive QL statements have been used for querying the data. Data visualization has been done by extracting the output of the HIVE query in excel and plotting the data using line and scatter plot charts. The visualization of the data shows some patterns that exist between flight diversions and flight distance, flight cancellation and flight distance and so forth. The U.S. Department of Transportation's (DOT) Bureau of Transportation Statistics (BTS) tracks the on-time performance of domestic flights operated by large air carriers. Summary information on the number of on-time, delayed, canceled, and diverted flights appear in DOT's monthly Air Travel Consumer Report, published about 30 days after the month's end, as well as in summary tables posted on this website. Summary statistics and raw data are made available to the public at the time the Air Travel Consumer Report is released.

Keywords: Hive, java, hadoop, Time Big Data Analytics, RTBDA, airline data320.

I. INTRODUCTION

Big data [1] is the term for a collection of data sets so large and complex. It becomes difficult to process using on-hand database management tools or traditional data processing applications. The challenges include capture, curation, storage, search, sharing, transfer, analysis and visualization. The trend to larger data sets is due to the additional information derivable from analysis of a single large set of related data, as compared to separate smaller sets with the same total amount of data, allowing correlations to be found to "spot business trends, determine quality of research, prevent diseases, link legal citations, combat crime, and determine real-time roadway traffic conditions.

As of 2012, limits on the size of data sets that are feasible to process in a reasonable amount of time were on the order of exabytes of data. Scientists regularly encounter limitations due to large data sets in many areas, including meteorology, genomics, connectomics, complex physics simulations, and biological and environmental research. The limitations also affect Internet search, finance and business informatics. Data sets grow in size in part because they are increasingly being gathered by ubiquitous information-sensing mobile devices, aerial sensory technologies (remote sensing), software logs, cameras, microphones, radio-frequency identification readers, and wireless sensor networks. The world's technological per-capita capacity to store information has roughly doubled every 40 months since the 1980s; as of 2012, every day 2.5 exabytes (2.5x1018) of data were created. The challenge for large enterprises is determining who should own big data initiatives that straddle the entire organization.

Big data is difficult to work with using most relational database management systems and desktop statistics and visualization packages, requiring instead "massively parallel software running on tens, hundreds, or even thousands of servers". What is considered "big data" varies depending on the capabilities of the organization managing the set, and on the capabilities of the applications that are traditionally used to process and analyze the data set in its domain. "For some organizations, facing hundreds of gigabytes of data for the first time may trigger a need to reconsider data management options. For others, it may take tens or hundreds of terabytes before data size becomes a significant consideration."

Hive [3] was created to make it possible for analysts with strong SQL skills (but meager Java programming skills) to run queries on the huge volumes of data airline stored in HDFS. Today, Hive is a successful Apache project used by many organizations as a general-purpose, scalable data processing platform

II. BIG DATA CHARACTERISTICS

In the airline data [4] most of the data is using in the online, almost every transaction is present in online. To understand the phenomenon of big data analytics we need to follow the five Vs as

Volume: Volume is a key contributor to the problem of why traditional relational database management systems. Volume always seems to head each list. There is general agreement that if volume is in the gigabytes it is probably not Big Data

Velocity: refers to the speed at which new data is generated and the speed at which data moves around. A second dimension of Velocity is how long the data will be valuable. Is it permanently valuable or does it rapidly age and lose its meaning and importance. Understanding this dimension of Velocity in the data you choose to store will be important in discarding data that is no longer meaningful and in fact may

www.rsisinternational.org

Page 177
mislead. Every day the number of emails, twitter messages, photos, video clips, etc.

Value: refers to our ability turn our data into value. It is important that businesses make a case for any attempt to collect and leverage big data. Value is that Big Data tends to have low value density. There are at least four additional characteristics that pop up in the literature from time to time. Big Data stores that they could not find in small stores. Certainly it is true that if in the past we were storing data about groups of customers an now storing data

Variety: refers to the different types of data we can now use. In the past we focused on structured data that neatly fits into tables or relational databases such as financial data. Variety describes different formats of data that do not lend themselves to storage in structured relational database systems. These include a long list of data such as documents, emails, social media text messages, video, still images, audio, graphs, and the output.

Veracity: refers to the messiness or trustworthiness of the data. With many forms of big data, quality and accuracy are less controllable. What is the provenance of the data? Does it come from a reliable source? It is accurate and by extension, complete. Big Data Veracity refers to the biases, noise and abnormality in Data. Inderpal feel veracity in data analysis is the biggest challenge when comparing things like volume and veracity figure 1 shows the 5Vs as

Big Data is a term used to describe large collections of data (also known as datasets) that may be unstructured, and grow so large and quickly that it is difficult to manage with regular database or statistics tools. Then the term “Big Data” comes into picture. They are several types.

III. MASSIVE DATA PROCESSING WITH HIVE: FLIGHT HISTORY ANALYSIS

The analysis and extraction of large amounts of data[5], which is usually related to the relational databases realm, has always represented a big challenge. Hadoop, Hive and Cloud computing services come to the rescue, offering a low-cost effective solution for “Big Data” analysis.

In this post we will show an example of parallel data processing using Hive to analyze a 30GB database; each query will cost us only about 0.37$. We will use the “On-Time” database from “TranStats”, which contains information about all the flights in the USA from 2005 to 2015 spread out over a total of 116 million registers in 240 CSV files. We will process these data with Hadoop using Hive, a high-level interface which allows us to execute Map-Reduce jobs based on a sequence of SQL-like commands. Thanks to this, we will perform a rapid, comprehensive analysis of these data in a simple, scalable and affordable way. At the end of the post we will specify the time and cost involved in executing this analysis.

To execute this analysis we need to create a Hadoop cluster with Hive, for instance in Amazon AWS. This task is easier to do if we use tools like Apache Whirr or Elastic MapReduce. While the first option is more technical, it’ll give us more freedom and lower costs after we’ve learned to use it. After creating the cluster, we can mount an EBS volume with the mentioned dataset and copy the necessary data to the HDFS.

Hadoop: Apache Hadoop is an open-source software framework for storage[7] and large scale processing of datasets on clusters of commodity hardware. Hadoop is an Apache top-level project being built and used by a global community of contributors and users.

HDFS: The Hadoop Distributed File System (HDFS) is a sub-project of the Apache Hadoop project. This Apache Software Foundation project is designed to provide a fault-tolerant file system designed to run on commodity hardware.

CLI Shell: Hive CLI (Command Line Interface), which is nothing but Hive Shell is the default service in Hive and it is the most common way of interacting with Hive.

Thrift Service: Hive Thrift Service is a software framework for scalable cross-language services development. Thrift allows clients using languages including Java, C++, Ruby, and many others, to programatically access Hive remotely.

HiveQL: Hive’s SQL dialect, called HiveQL, does not support the full SQL-92 specification. There are a number of reasons for this. Being a fairly young project, it has not had time to provide the full repertoire of SQL-92 language constructs. More fundamentally, SQL-92 compliance has never been an explicit project goal; rather, as an open source project, features were added by developers to meet their users’ needs.

Furthermore, Hive has some extensions that are not in SQL-92, which have been inspired by syntax from other database systems, notably MySQL. In fact, to a first-order approximation, HiveQL most closely resembles MySQL’s SQL dialect.

Hive Shell: The Hive Shell The shell is the primary way that we will interact with Hive, by issuing commands in HiveQL.
HiveQL is Hive’s query language, a dialect of SQL. It is heavily influenced by MySQL, so if you are familiar with MySQL, you should feel at home using Hive. When starting Hive for the first time, we can check that it is working by listing its tables — there should be none. The command must be terminated with a semicolon to tell Hive to execute it:

```sql
hive> SHOW TABLES;
OK
Time taken: 0.473 seconds
```

Like SQL, HiveQL is generally case insensitive (except for string comparisons), so show tables; works equally well here. The Tab key wills auto complete Hive keywords and functions. For a fresh install, the command takes a few seconds to run as it lazily creates the metastore database on your machine. (The database stores its files in a directory called metastore_db, which is relative to the location from which you ran the hive command.)

**Hive Services**: The Hive shell is only one of several services that you can run using the hive command. You can specify the service to run using the --service option. Type hive --service help to get a list of available service names; some of the most useful ones are described in the following list:

- **cli**: The command-line interface to Hive (the shell). This is the default service.
- **Hiveserver2**: Runs Hive as a server exposing a Thrift service, enabling access from a range of clients written in different languages. HiveServer2 improves on the original HiveServer by supporting authentication and multiuser concurrency. Applications using the Thrift, JDBC, and ODBC connectors need to run a Hive server to communicate with Hive. Set the hive.server2.thrift.port configuration property to specify the port the server will listen on (defaults to 10000).
- **Beeline**: A command-line interface to Hive that works in embedded mode (like the regular CLI), or by connecting to a HiveServer2 process using JDBC.
- **Hwi**: The Hive Web Interface. A simple web interface that can be used as an alternative to the CLI without having to install any client software. See also Hue for a more fully featured Hadoop web interface that includes applications for running Hive queries and browsing the Hive metastore.
- **Jar**: The Hive equivalent of hadoop jar, a convenient way to run Java applications that includes both Hadoop and Hive classes on the classpath.
- **Metastore**: By default, the metastore is run in the same process as the Hive service. Using this service, it is possible to run the metastore as a standalone (remote) process. Set the METASTORE_PORT environment variable (or use the -p command-line option) to specify the port the server will listen on (defaults to 9083).

**Hive Clients**: Hive clients If you run Hive as a server (hive --service hiveserver2), there are a number of different mechanisms for connecting to it from applications.

**Thrift Client**: The Hive server is exposed as a Thrift service, so it’s possible to interact with it using any programming language that supports Thrift.

**IV. HIVE**

Apache Hive[8] is a Data Warehousing package built on top of Hadoop and is used for data analysis. Hive is targeted towards users who are comfortable with SQL. It is similar to SQL and called HiveQL, used for managing and querying structured data. Apache Hive is used to abstract complexity of Hadoop. This language also allows traditional map/reduce programmers to plug in their custom mappers and reducers. The popular feature of Hive is that there is no need to learn Java.

Hive, an open source peta-byte scale date warehousing framework based on Hadoop, was developed by the Data Infrastructure Team at Facebook. Hive is also one of the technologies that are being used to address the requirements at Facebook. Hive is very popular with all the users internally at Facebook and is being used to run thousands of jobs on the cluster with hundreds of users, for a wide variety of applications. Hive-Hadoop cluster at Facebook stores more than 2PB of raw data and regularly loads 15 TB of data on a daily basis.

**Hive Applications:**

Fig:Hive applications

**MetaStore**: Hive, metastore is the central repository to store metadata for hive tables/partitions. Any datastore that has a JDBC driver can be used as a metastore. By default the metastore service runs in the same JVM. Hive service and contains embedded Derby database instance backed by the local disk. There are 3 different ways to setup the metastore server using different Hive configurations:

**Embedded Metastore**: By default, the Metastore service runs in the same JVM with the hive service. In this case it uses embedded derby database stored on the local file-system. To
allow multiple Hive services to connect the Metastore, Derby is configured as a network server.

**Local Metastore:** Being a data-warehousing framework, a single session for Hive is not preferred. To solve this limitation of Embedded Metastore.A support for Local Metastore was developed. A separate database service runs as a process on same or remote machine. The Metastore service still runs in the same JVM within hive service. Before starting a Hive client, add the JDBC / ODBC driver libraries to the Hive lib folder.

**Remote Metastore:** There is one more configuration where one or more Metastore servers run as separate processes. This allows multiple Hive Clients to connect to a remote service rather than starting a Metastore service in the same JVM.

### 4.1 Hive Architecture

![Hive Architecture](image)

**Thrift Client:** The Hive Thrift Client[14] makes it easy to run Hive commands from a wide range of programming languages. Thrift bindings for Hive are available for C++, Java, PHP, Python, and Ruby. They can be found in the `src/service/src` subdirectory in the Hive distribution.

**JDBC Driver:** Hive provides a Type 4 (pure Java) JDBC driver, defined in the class `org.apache.hadoop.hive.jdbc.HiveDriver`. When configured with a JDBC URI of the form `jdbc:hive://host:port/dbname`, a Java application will connect to a Hive server running in a separate process at the given host and port. (The driver makes calls to an interface implemented by the Hive Thrift Client using the Java Thrift bindings.) At the time of writing, default is the only database name supported. You may alternatively choose to connect to Hive via JDBC in embedded mode using the URI `jdbc:hive://`. In this mode, Hive runs in the same JVM as the application invoking it; there is no need to launch it as a standalone server, since it does not use the Thrift service or the Hive Thrift Client.

The JDBC driver is still in development, and in particular it does not support the full JDBC API.

**ODBC Driver:** The Hive ODBC Driver allows applications that support the ODBC protocol to connect to Hive. (Like the JDBC driver, the ODBC driver uses Thrift to communicate with the Hive server.) The ODBC driver is still in development, so you should refer to the latest instructions on the Hive wiki for how to build and run it.

### V. HIVE PROCESS THROUGH JAVA

Initially the language was called as “oak” but it was renamed as “java” in 1995. The primary motivation of this language was the need for a platform-independent(i.e. architecture neutral) language that could be used to create software to be embedded in various consumer electronic devices.

- Java is a programmer’s language
- Java is cohesive and consistent

Except for those constraint imposed by the Internet environment. Java gives the programmer, full control. Finally Java is to Internet Programming where c was to System Programming.

**JDBC driver:** Hive provides a Type 4 (pure Java) JDBC driver, defined in the class `org.apache.hadoop.hive.jdbc.HiveDriver`. When configured with a JDBC URI of the form `jdbc:hive2://host:port/dbname`, a Java application will connect to a Hive server running in a separate process at the given host and port. (The driver makes calls to an interface implemented by the Hive Thrift Client using the Java Thrift bindings.) You may alternatively choose to connect to Hive via JDBC in embedded mode using the URI `jdbc:hive2://`. In this mode, Hive runs in the same JVM as the application invoking it; there is no need to launch it as a standalone server, since it does not use the Thrift service or the Hive Thrift Client. The Beeline CLI uses the JDBC driver to communicate with Hive.

**ODBC driver:** An ODBC driver allows applications that support the ODBC protocol (such as business intelligence software) to connect to Hive. The Apache Hive distribution does not ship with an ODBC driver, but several vendors make one freely available. (Like the JDBC driver, ODBC drivers use Thrift to communicate with the Hive server.)

**Hive Tables:** A Hive table is logically made up of the data being stored and the associated metadata describing the layout of the data in the table. The data typically resides in HDFS, although it may reside in any Hadoop filesystem, including the local filesystem or S3. Hive stores the metadata in a relational database and not in, say, HDFS.

- Managed table
- External table

**Managed table:** Managed table is also known as internal table. It can be accessed by particular user locally. Whenever you create an internal table, the data in that table is moved into the warehouse directory. Both the data and schema are under

---

www.rsisinternational.org  Page 180
the control of hive. On dropping these tables, the data gets deleted permanently.

**External table:** External table can be accessed by all the users. Whenever external table is created by the user, the data in that gets stored at an existing location outside the warehouse directory. It is mostly preferred as data is stored in HDFS. Only schema is under the control of hive. On dropping these tables, only metadata gets deleted and actual data remains the same.

**Creation of table:** Create Table is a statement used to create a table in Hive.

create table studentdatabase(student_id bigint, student_name string, year bigint, branch string, section string, date string, time stamp)
row format delimited
Fields terminated by ‘,’

**Loading data from HDFS to Hive:** Load data inpath /user/training/studentdb into table studentdatabase;

**Java Architecture:** Java architecture provides a portable, robust, high performing environment for development. Java provides portability by compiling the byte codes for the Java Virtual Machine, which is then interpreted on each platform by the run-time environment. Java is a dynamic system, able to load code when needed from a machine in the same room or across the planet.

**Compilation of code:** When you compile the code, the Java compiler creates machine code (called byte code) for a hypothetical machine called Java Virtual Machine (JVM). The JVM is supposed to execute the byte code. The JVM is created for the overcoming the issue of probability. The code is written and compiled for one machine and interpreted on all machines. This machine is called Java Virtual Machine.

**Compiling and interpreting java source code.**

During run-time the Java interpreter tricks the byte code file into thinking that it is running on a Java Virtual Machine. In reality this could be an Intel Pentium windows 95 or sun SPARCstation running Solaris or Apple Macintosh running system and all could receive code from any computer through internet and run the Applets. In the previous days system is marinating the information of databases in very high cost. The database itself maintains in limited memory only. To rectify this problems by using big data Hadoop framework. We can see that Interesting sets of trends and patterns exists in large data sets which helps us to get a better understanding of data. Here author focused on the processing the big data sets using hive component of hadoop ecosystem in distributed environment. This work will benefit the developers and business analysts in accessing and processing their use quires

**VI. CONCLUSION**

We have processed 40 GB of data in different ways: we have extracted reports, executed “ad-hoc” queries, generated graphs and detected outliers in them in a simple and effective way with Hive. All these processes scale horizontally because they use Hadoop underneath, so if our data was double the size, we would only need to double the number of machines in our cluster to keep executing the same processes in approximately the same time. Without needing to use proprietary software or hardware, we can have the power of parallel data processing at an affordable price thanks to these new technologies.

**REFERENCES**

[1]. Bernard Marr (2015), Big data and Big data in practices, 2nd edition, Willy publication.
[4]. Airline Data Set, United States Department of Transportation, Office of the Assistant Secretary for Research and Technology, Bureau of Transportation Statistics, http://www.transtats.bts.gov/DL_SelectFields.asp?Table_ID=236
[8]. MapR, the Next Generation Distribution for Apache Hadoop http://mapr.com


[16]. https://cwiki.apache.org/Hive/

[17]. JDK 1.6 java.util.regex.Pattern http://docs.oracle.com/javase/6/docs/api/java/util/regex/Pattern.htm