Literature Review

Team Members:

Big Data refers to the ways to analyze data, systematic extraction of information, and deal with the dataset that is too large to deal with using the traditional software. The term was first referred to in the year 2005 by Roger Mougalas from O'Reilly Media. It is referred to as a large dataset that was merely impossible to manage and process. The measurement of the big data is not fixed to one size; in some cases, 2TB can be big data, and in another case, 200 TB can also be big data. In other words, "Big Data is a circumstance where the volume, velocity, and variety of data go beyond an organization's storage or computation capacity for precise and well-timed decision making". Big Data is characterized mainly into Five terms of Vs.

Volume: it refers to the sheer size of the data. These datasets can be orders of magnitude larger than the traditional datasets. With an increase in the growth of social media, the data generated is also growing exponentially. The data generated through machines exceeds the data generated by humans.

Velocity: Velocity is the speed at which the data is generated and moves through the system. Data is frequently flowing into the system from multiple sources and is often expected to be processed in real-time. With a focus on instant feedback and instant solutions has made the developers shift from batch oriented-approach to real-time streaming system.

Variety: Variety refers to the format in which the data is generated. Be it structured, unstructured, or semi-structured data, but 70% of the data generated is unstructured data. In traditional days the information was structured like spreadsheets, databases, flat files, etc., and nowadays the share of structured data is too low, the unstructured data that today is generated are in the form of video files, images, weblogs, sensor data, audio clips, etc.

Veracity: Veracity is the fourth V in the 5 V's of big data. It refers to the quality and accuracy of data. Gathered data could have missing pieces, may be inaccurate or may not be able to provide real, valuable insight. Veracity, overall, refers to the level of trust there is in the collected data.

Data can sometimes become messy and difficult to use. A large amount of data can cause more confusion than insights if it's incomplete. For example, concerning the medical field, if data about what drugs a patient is taking is incomplete, then the patient's life may be endangered.

Value: This refers to the value that big data can provide, and it relates directly to what organizations can do with that collected data. Being able to pull value from big data is a

requirement, as the value of big data increases significantly depending on the insights that can be gained from them.

Organizations can use the same big data tools to gather and analyze the data, but how they derive value from that data should be unique to them.

Even though RDBMS is the most preferred tool in IT, it failed when it comes to Big Data. One of the reasons that it failed for Big Data was that RDBMS could not handle outsized data with a

variety.

RDBMS follows a very strict schema with lots of constraints for the data. As a fact, it is known that most of the data in Big Data is in an unstructured format; it will always be challenging to have a schema. And maintaining relationships for unstructured data (video, weblogs, images, audio clips, etc.) is almost impossible. For analyzing a small dataset, time taken to process can be neglected, but for extensive datasets, time is a significant factor. Big data should possess fast processing speed like real-time insights, which RDBMS can't. Processing Big Data with traditional methods would not be cost-effective and a time-consuming process; to overcome these drawbacks, new technology was introduced called Hadoop.

Hadoop has subprojects like Hive, Pig, Spark Kafka, HBase, Oozie, which are an excellent option for Big Data Analytics. Most of the Big Data technologies are open-source software and can be used by anyone; some vendors, on the other hand, enhance this software to a better version with paid services. Hadoop is written in Java and can run on commodity hardware, scaling up from a single node to thousands of computers, thus creating a massive cluster.

Apache Pig

In 2006, Apache Pig was developed by Yahoo's research team. Apache Pig helps in processing large datasets. The programmers will use the Pig Latin Language to process the data which is stored in the HDFS. Internally, Pig Engine converts all these scripts into a specific map and reduces tasks. Pig Latin and Pig Engine are the two main components of the Apache Pig tool, which outputs the required results that are always stored in the HDFS. It is an interactive execution environment which uses PigLatin, unlike in Hive, relations are expressed as data flows.The flow in Pig starts with checking the syntax of the script and gives an output in the form of a DAG (Directed Acyclic Graph). Then, DAG is passed to the logical optimizer with the help of parser. It carries out optimizations such as projections and pushdowns. It is then transferred to the compiler, which compiles the optimized DAG into a series of Map-Reduce jobs, which then gives the final output once map-reduce jobs are run. Pig can be run in three different ways; all of them are compatible with local and Hadoop:

Script: Simply a file containing Pig Latin commands, identified by the .pig suffix. Pig interpret the commands and executes in sequential order.

Grunt: Grunt is a command interpreter. It can be typed in Pig Latin on the grunt command line, and Grunt will execute the command. It is beneficial for prototyping and "what if" scenarios.

Embedded: Pig programs can be executed as part of a java program.

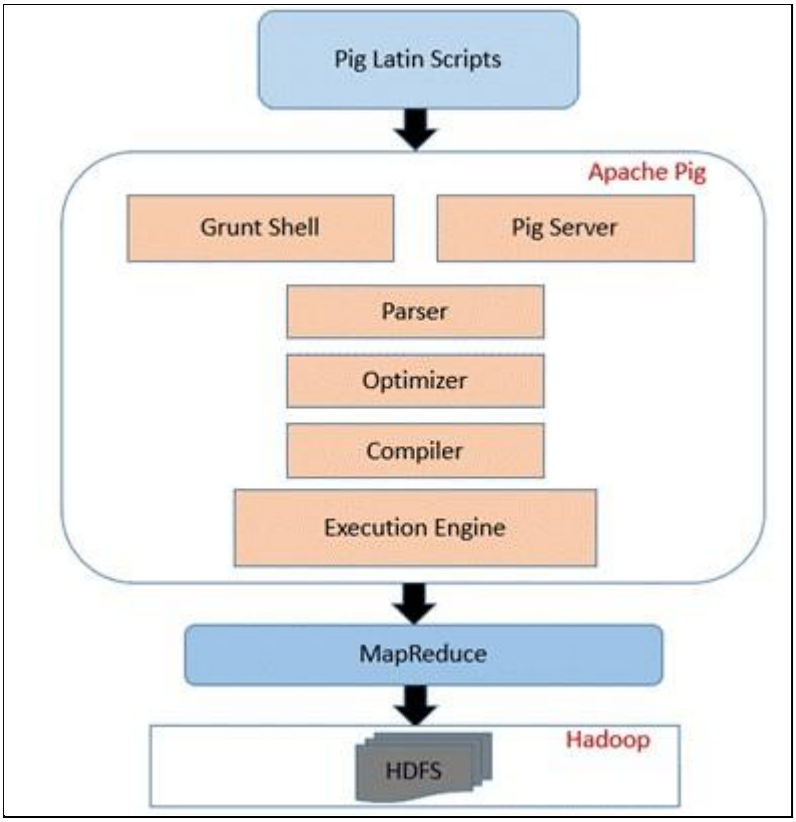
Workflow of Apache Pig

Parser: It checks the script for syntax. The output of this component will be a DAG – Directed Acyclic Graph, representing PigLatin statements.

Optimizer: The DAG made by the parser is passed to the optimizer which carries out logical optimization like projection and pushdown.

Compiler: It compiles the optimized plan into MapReduce jobs.

Execution Engine: In the final stage, the MapReduce jobs are submitted to Hadoop in a sorted manner and are then executed to derive the desired output.



Apache Hive

Apache Hive is an open-source data warehouse tool which is useful for analyzing large data sets. Apache Hive uses a query language called Hive query language, which supports ACID properties in HiveQL, with SQL commands like the update, insert, and delete. Hive query written in the command-line interface is delivered to the driver. The driver creates a session of the handle and then transfers the question to the compiler. The compiler assigns the metadata request to the database and extracts the required information. The compiler finally prepares an execution plan and shares it with the driver; subsequently, the result is transferred to the execution engine.

- Authors of the given research paper compare the efficiency of the MySQL server, Apache Hive, and Apache Pig. They have derived their conclusion based on the query statements and the average query time. They have used three datasets for their analysis: m1100k (movie lens 100,000 rows), m11m containing a total of 1,075,611 rows, and m110m containing a total of 10,069,372 rows.

- Pig executes using a step-by-step approach. Pig works well when a query has a

sophisticated type of function and many joins in the data, Pig can handle it efficiently by simultaneously executing each step and the subsequent next step. This approach does not work well in a query that has minimum joins and filters, as it can consume more time.

- Hive has an advantage, such as indexing, which leads to faster file reading [3]. Hive invokes MapReduce only if the query has aggregation, join, or sorting function, which can take one to six seconds to start the MapReduce.

- Hive is capable of handling extensive data and is faster than MySQL. Pig is not suitable for such a data set. Pig is more ideal for more complex queries and more massive data sets.

Hive Components:

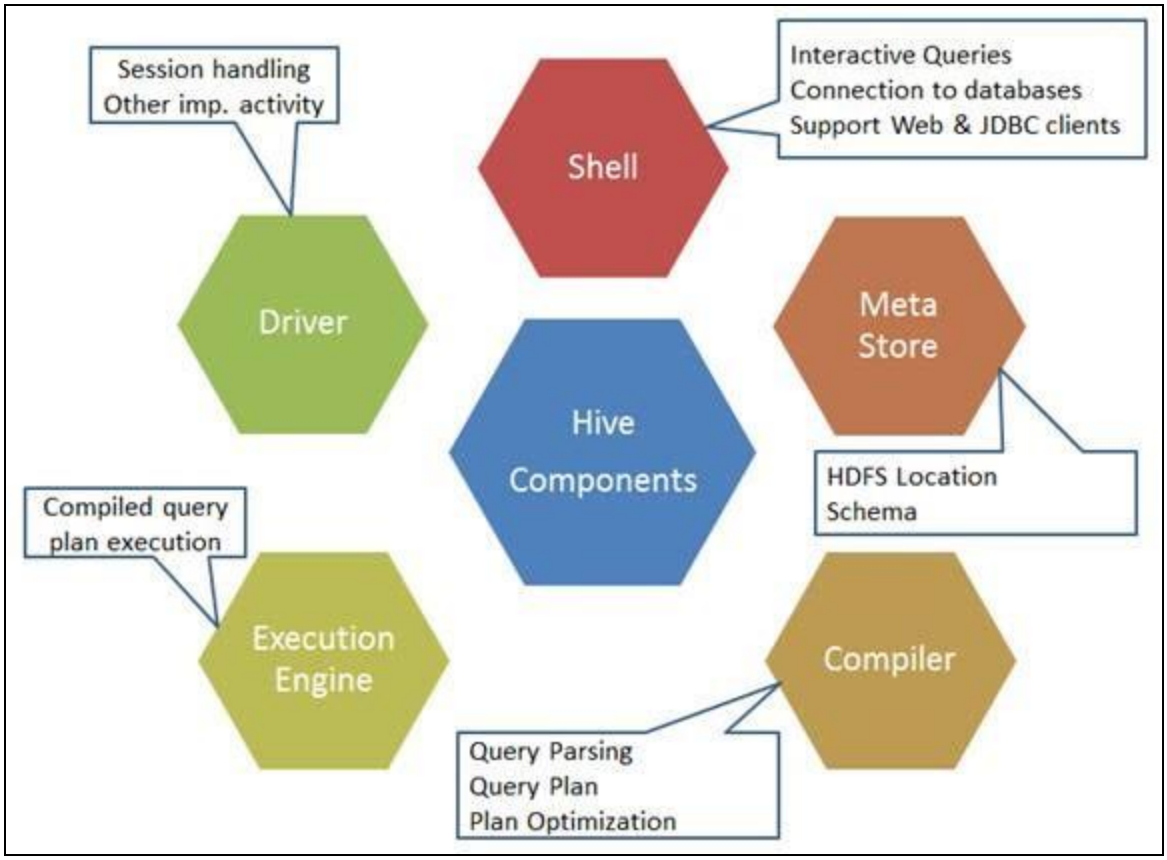
Metastore: A repository for metadata of Hive. It consists of information like data location, schema along with metadata of partitions.

Compiler: User for compiling Hive queries into map-reduce jobs and running them.

Driver: A controller mainly responsible for storing the generated metadata while executing HQL statements.

Optimizer: It splits tasks during the execution of map- reduce jobs, thus helping in scalability and efficiency.

Hive Shell: A terminal user for interacting with Hive to run Hive queries and is not case sensitive.



Apache Spark(PySpark)

Apache Spark is a distributed general-purpose cluster computing network, which has an interface for entire programming clusters with implicit data parallelism and fault tolerance. Spark has RDD – Resilient Distributed Dataset as an architectural foundation, which is a read-only multiset of data items distributed over a cluster of machines in a fault-tolerant manner. The spark came in as an alternative for map-reduces' limitations, a cluster computing paradigm that forces a linear data flow structure on distributed programs. Apache Spark has a different approach; it implements both iterative algorithms, which visit data set multiple times in a loop and interactive data analysis which is the repetition of database styled query of data. Spark Core is the overall foundation of the project "Apache Spark," it provides distributed task dispatching, scheduling, and basic input/output functionalities through an interface (Python, Scala R, etc.) centered on the RDD abstraction. In previous map-reduce jobs, the workloads required separate engines, including SQL, streaming, graph processing, machine learning, but RDD in Spark handles all these workloads and works as a single-engine for all requirements. These implementations use the same optimizations as specialized engines like column-oriented processing and incremental updates; and achieve similar performance but run libraries over a common engine, making it easy and efficient to compose. A few of the benefits that Spark gives are that applications are

more comfortable to develop as they use a unified API; secondly it is more suited to combine processing tasks. Third, Spark can run diverse functions over the same data, often in memory. As parallel data processing is becoming common, the composability of processing functions is also becoming an essential concern in terms of usability and performance. RDD is lazily evaluated by Spark to find an efficient plan for user computation.

When an action is called, Spark looks at the whole graph of transformations used to create an execution plan. For example, consider if there were multiple filters or map operations in a row, Spark fuses them into one pass, or it is known to spark in technical language as data is partitioned. It avoids it over the network for groupby. Thus, users can build programs modularly without losing performance.

One of the options for performing Big Data Analytics in Spark is PySpark. PySpark is the combination of Apache Spark and Python. Deep learning has been in the limelight for quite some time in the market. Challenges in Deep Learning are exponentially rising as use of Deep Learning is taking gigantic leaps in numerous real business scenarios. It can't be denied that many corporations are dependent heavily on deep learning like image language translation, self-driving cars to drone deliveries. Google is one of the few companies who have completely engulfed Deep Learning in day-to-day operations (Gmail, YouTube, Maps, Chrome, Google Assistance, Google Translation etc.). PySpark is a Python based API for Spark which enables users to use the functionalities of Spark with the power of Python libraries. Data in PySpark can be stored and accessed using RDD (Resilient Distributed Dataset) and Spark DataFrame. RDD are bases of Spark, it is fault tolerant and the data can be distributed among various nodes in a cluster. On the other hand, unlike Pandas DataFrame, Spark DataFrame is a distributed collection of structured and semi-structured data. Its functionalities are analogous to relational database tables. It can either be loaded from existing RDD or creating a new schema.

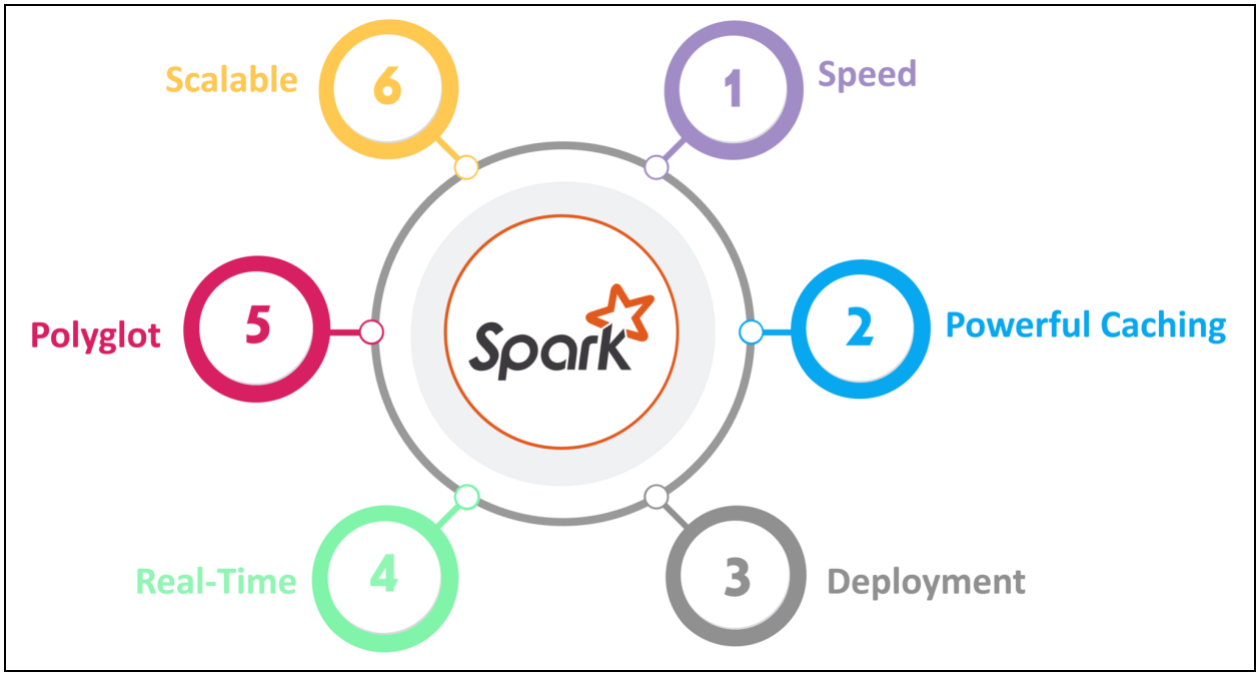
PySpark Features

Speed: It's nearly 100 times faster than the traditional large-scale data processing.

Powerful Caching: Simple programming layer in PySpark provides powerful caching and disk persistence capabilities.
Deployment: PySpark can be deployed through Mesos, Hadoop via Yarn, or Spark's own cluster manager.

Real-Time: Real-time computation and low latency because of in-memory computation.

Polyglot: Supports programming in Scala, Java, Python, and R.



Google Cloud Platform (Big Query)

Big Query is Google's data warehouse which is managed at petabyte scale and at a very low cost. Big Query follows NoOps structure – which means that there is no infrastructure to manage and one doesn't need a database administrator. It follows SQL-querying systems for fetching datasets.

The main feature of the Google Big Query are as follows:

- **Managing Data**: Data can be pulled in the csv or json format.
- **Query:** The queries for extracting the datasets are expressed in the SQL dialect.
- **Integration**:The Big Query can be used from the Google app script.
- **Access Control**:The datasets can be shared with other users.

Team Members Responsibilities:

Pooja Pramod Kantrod - Gathering the required csv's and data files from Google's Big Query and also helping to write the queries

Raj Banker - Setting up the Pig environment and executing the Pig commands and writing the queries.

Naga Surya Suresh Lnu - Setting up the Hive environment and Executing the Hive Commands and writing the queries.

Rahul Maddula - Setting up the PySpark environment to build a prediction model ,Document Preparation(Project proposal, Project Draft, Project Report) and also assisting to write the queries.

References:

A. Fuad, A. Erwin, and H. Ipung, "Processing performance on Apache Pig, Apache Hive and MySQL cluster," IEEE, no. 10110920147010600, 2019. Available: 10.1109/ICTS.2014.7010600 [Accessed 17 October 2019].

"Introduction to Apache Hive | Edureka.co", Edureka, 2019. [Online]. Available: https://www.edureka.co/blog/introduction-to-apache-hive/comment-page-1/ . [Accessed: 24Nov-2019].

J. Hurwitz, A. Nugent, F. Halper, and M. Kaufman, "Hadoop Pig and Pig Latin for Big Data dummies," dummies, 2019. [Online]. Available: https://www.dummies.com/programming/big-data/hadoop/hadoop-pig-and-pig-latin-for-bigdata/. [Accessed: 22- Nov- 2019].

Lecture Slides.