Apache Pig

WHAT IS HADOOP PIG/ APACHE PIG?

- Hadoop Pig is nothing but an abstraction over MapReduce.
- While it comes to analyzing large sets of data, as well as to represent them as data flows, we use Apache Pig.
- Generally, we use it with Hadoop. By using Pig, we can perform all the data manipulation operations in Hadoop.
- In addition, Pig offers a high-level language to write data analysis programs which we call as **Pig Latin**.
- One of the major advantages of this language is it offers several operators. Through them, programmers can develop their own functions for reading, writing, and processing data.

KEY PROPERTIES OF PIG

- **Ease of programming**: Basically, when all the complex tasks comprised of multiple interrelated data transformations are explicitly encoded as data flow sequences, that makes them easy to write, understand, and maintain.
- **Optimization opportunities**: It allows users to focus on semantics rather than efficiency, to optimize their execution automatically, in which tasks are encoded permits the system.
- **Extensibility**: In order to do special-purpose processing, users can create their own functions.

Hence, programmers need to write scripts using Pig Latin language to analyze data using Apache Pig.

- All the scripts are internally converted to Map and Reduce tasks. It is possible with a component, we call it as Pig Engine.
- That accepts the Pig Latin scripts as input and further convert those scripts into MapReduce jobs.
- Apache Pig enables people to focus more on analyzing bulk data sets and to spend less time writing Map-Reduce programs.
- Similar to Pigs, who eat anything, the Apache Pig programming language is designed to work upon any kind of data. That's why the name, Pig!

HISTORY

- Apache Pig was developed as a research project, in 2006, at Yahoo.
- Basically, to create and execute MapReduce jobs on every dataset it was created.
- By Apache incubator, Pig was open sourced in 2007.
- Then the first release of Apache Pig came out in 2008.
- Further, Hadoop Pig graduated as an Apache top-level project, in 2010.

WHY DO WE NEED APACHE PIG?

- While performing any MapReduce tasks, there is a case Programmers who are not so good at Java normally used to struggle to work with Hadoop.
- Without having to type complex codes in Java, using Pig Latin, programmers can perform MapReduce tasks easily.
- It also helps in reduce the length of codes, since Pig uses multi-query approach.
- When you are familiar with SQL, it is easy to learn Pig Latin.
- It offers many built-in operators, in order to support data operations such as joins, filters, ordering, and many more.
- Also, it offers nested data types that are missing from MapReduce such as tuples, bags, and maps.

USING PIG

- While data loads are time sensitive.
- While processing various data sources.
- While we require analytical insights through sampling.

WHERE NOT TO USE PIG?

- While the data is completely unstructured. Such as video, audio, and readable text.
- Where time constraints exist since Pig is **slower** than MapReduce jobs.
- Also, when more power is required to optimize the codes, we cannot use Pig.

EXECUTION MODES

- Pig in Hadoop has two execution modes:
- **Local mode:** In this mode, Hadoop Pig language runs in a single JVM and makes use of local file system. This mode is suitable only for analysis of small datasets using Pig in Hadoop

```
pig –x local
```

- **Map Reduce mode**: In this mode, queries written in Pig Latin are translated into MapReduce jobs and are run on a Hadoop cluster (cluster may be pseudo or fully distributed). MapReduce mode with the fully distributed cluster is useful of running Pig on large datasets.

```
pig –x mapreduce
```

OR `pig`

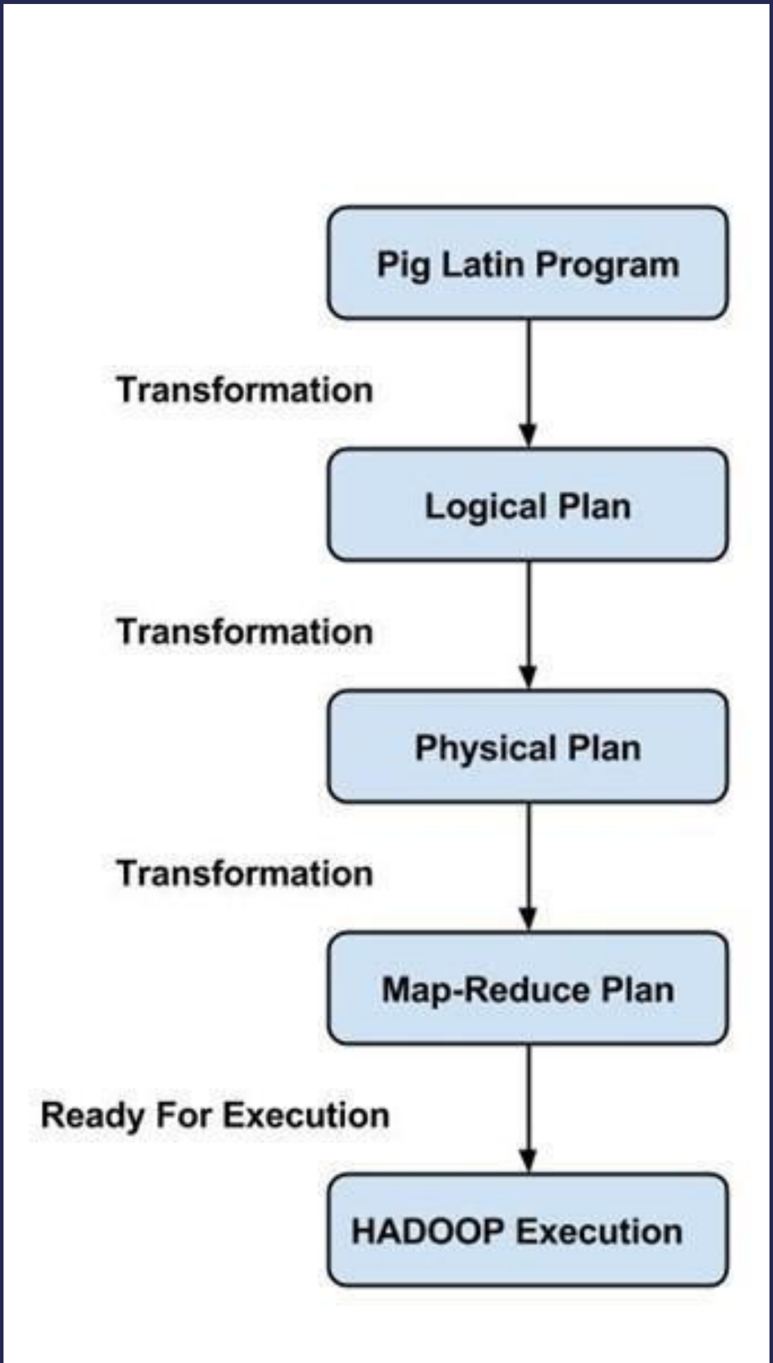
ARCHITECTURE OF HADOOP PIG

- The Architecture of Pig consists of two components:

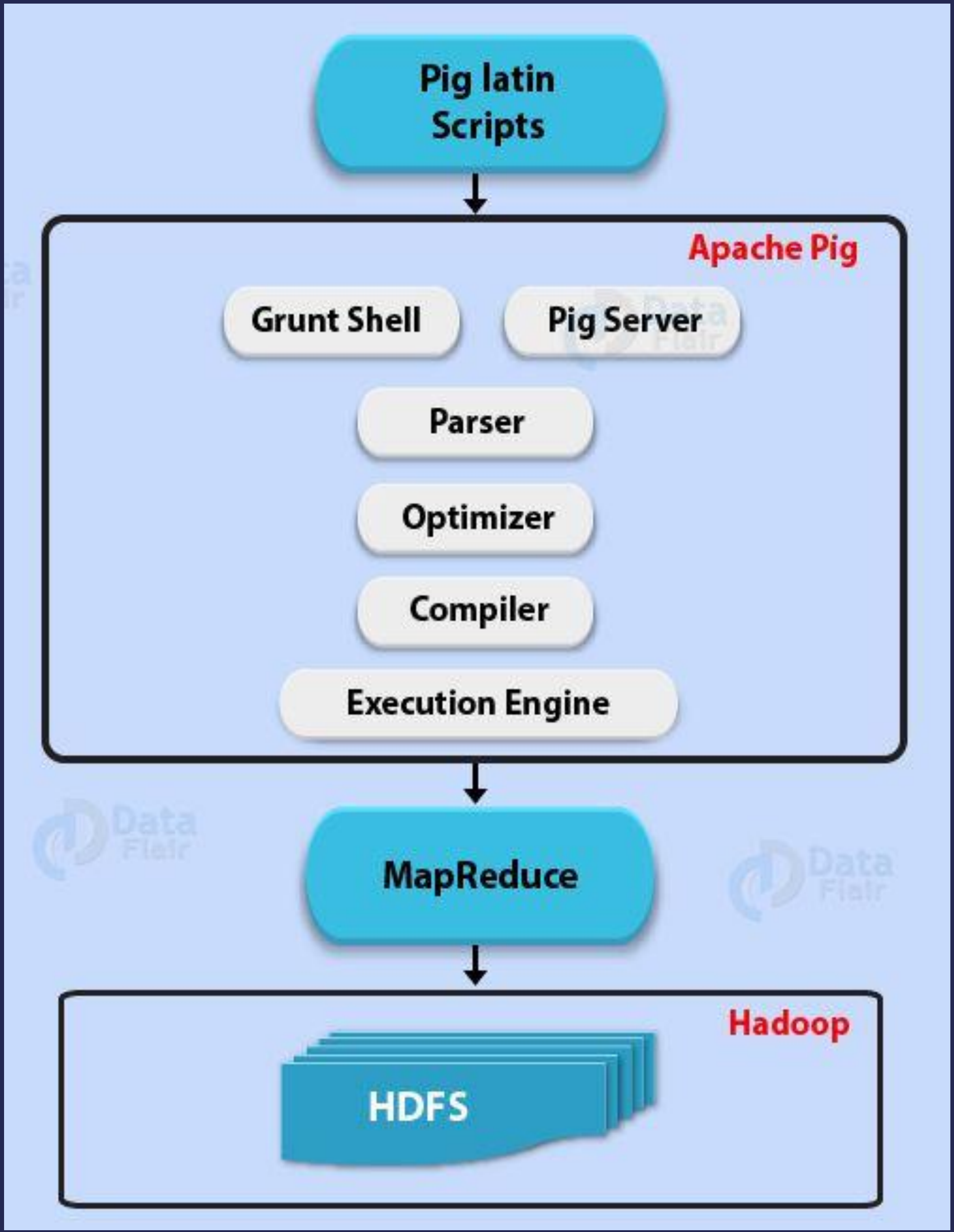
1. **Pig Latin**, which is a language
2. A runtime environment**, Pig Engine**, for running PigLatin programs.

- A Pig Latin program consists of a series of operations or transformations which are applied to the input data to produce output.
- These operations describe a data flow which is translated into an executable representation, by Hadoop Pig execution environment. Underneath, results of these transformations are series of MapReduce jobs which a programmer is unaware of.

- Pig in Hadoop allows the programmer to focus on data rather than the nature of execution.
- Pig Latin is a relatively stiffened language which uses familiar keywords from data processing e.g., Join, Group and Filter.



Pig Latin Program
Transformation
Logical Plan
Transformation
Physical Plan
Transformation
Map-Reduce Plan
Ready For Execution
HADOOP Execution



Pig latin Scripts
Apache Pig
Grunt Shell
Pig Server
Parser
Optimizer
Compiler
Execution Engine
MapReduce
Hadoop
HDFS

I. PARSER

- At first, all the Pig Scripts are handled by the Parser.
- Basically, Parser checks the syntax of the script, does type checking, and other miscellaneous checks.
- Afterward, Parser's output will be a DAG (directed acyclic graph). That represents the Pig Latin statements as well as logical operators.
- Basically, the logical operators of the script are represented as the nodes and the data flows are represented as edges, in the DAG (the logical plan).

II. OPTIMIZER

- Further, DAG is passed to the logical optimizer. That carries out the logical optimizations. Like projection and push down.

III. COMPILER

- It compiles the optimized logical plan into a series of MapReduce jobs.

IV. EXECUTION ENGINE

- At last, MapReduce jobs are submitted to Hadoop in a sorted order. Hence, these MapReduce jobs are executed finally on Hadoop, that produces the desired results.

PIG FEATURES

- **Rich set of operators**: In order to perform several operations, Pig offers many operators. Such as join, sort, filer and many more.
- **Ease of programming**: Since you are good at SQL, it is easy to write a Pig script. Because of Pig Latin as same as SQL.
- **Optimization opportunities**: In Apache Pig, all the tasks optimize their execution automatically. As a result, the programmers need to focus only on the semantics of the language.

- **Extensibility**: Through Pig, it is easy to read, process, and write data. It is possible by using the existing operators. Also, users can develop their own functions.
- **UDFs**: By using Pig, we can create User-defined Functions in other programming languages like Java. Also, can invoke or embed them in Pig Scripts.
- **Handles all kinds of data**: Pig generally analyzes all kinds of data. Even both structured and unstructured. Moreover, it stores the results in HDFS.