Unit 4

Pig: Hadoop programming Made Easier

Apache Pig is an abstraction over MapReduce. It is a tool/platform which is used to analyze larger sets of data representing them as data flows. Pig is generally used with **Hadoop**; we canperform all the data manipulation operations in Hadoop using Pig.

To write data analysis programs, Pig provides a high-level language known as **Pig Latin**. This language provides various operators using which programmers can develop their functionsfor reading, writing, and processing data.

To analyze data using **Apache Pig**, programmers need to write scripts using Pig Latin language. All these scripts are internally converted to Map and Reduce tasks. Apache Pig has a componentknown as **Pig Engine** that accepts the Pig Latin scripts as input and converts those scripts into MapReduce jobs.

Why Do We Need Apache Pig?

Programmers who are not so good at Java normally used to struggle to work with Hadoop,especially while performing any MapReduce tasks. Apache Pig is a boon for all such programmers.

- Using Pig Latin, programmers can perform MapReduce tasks easily without having totype complex codes in Java.
- Apache Pig uses a multi-query approach, thereby reducing the length of codes. For example, an operation that would require you to type 200 lines of code (LoC) in Java canbe easily done by typing less than just 10 LoC in Apache Pig. Ultimately Apache Pig reduces the development time by almost 16 times.
- Pig Latin is an SQL-like language and it is easy to learn Apache Pig when you are familiarwith SQL.
- Apache Pig provides many built-in operators to support data operations like joins, filters, ordering, etc. In addition, it also provides nested data types like tuples, bags, and maps that are missing from MapReduce.

Features of Pig

Apache Pig comes with the following features −

- **Rich set of operators** − It provides many operators to perform operations like join, sort,filer, etc.
- **Ease of programming** − Pig Latin is similar to SQL and it is easy to write a Pig script ifyou are good at SQL.
- **Optimization opportunities** − The tasks in Apache Pig optimize their execution automatically, so the programmers need to focus only on the semantics of the language.
- **Extensibility** − Using the existing operators, users can develop their functions toread, process, and write data.
- **UDF's** − Pig provides the facility to create User-defined Functions in other programminglanguages such as Java and invoke or embed them in Pig Scripts.
- **Handles all kinds of data** − Apache Pig analyzes all kinds of data, both structured aswell as unstructured. It stores the results in HDFS.

Apache Pig Vs MapReduce

Listed below are the major differences between Apache Pig and MapReduce.

Apache Pig	MapReduce
Apache Pig is a data flow language.	MapReduce is a data processingparadigm.
It is a high-level language.	MapReduce is low level and rigid.
Performing a Join operation in Apache Pig ispretty simple.	It is quite difficult for MapReduce to perform a Join operation between datasets.
Any novice programmer with a basic knowledge of SQL can work conveniently with Apache Pig.	Exposure to Java is a must towork with MapReduce.
Apache Pig uses a multi-query approach, thereby reducing the length of the codes toa great extent.	MapReduce will require almost 20 times more the number of lines toperform the same task.
There is no need for compilation. On execution, every Apache Pig operator is converted internally into a MapReducejob.	MapReduce jobs have a long compilation process.

Apache Pig Vs SQL

Listed below are the major differences between Apache Pig and SQL.

Pig	SQL
Pig Latin is a procedural language.	SQL is a declarative language.
In Apache Pig, the schema is optional. We can store data without designing a schema (valuesare stored as $01, $02, etc.)	Schema is mandatory in SQL.
The data model in Apache Pig is nestedrelational.	The data model used in SQL isflat relational.
Apache Pig provides limited opportunityfor Query optimization.	There is more opportunity for query optimization in SQL.

In addition to the above differences, Apache Pig Latin −

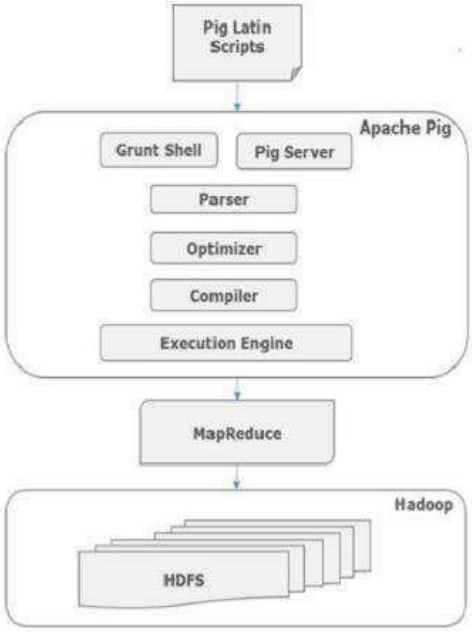
- Allows splits in the pipeline.
- Allows developers to store data anywhere in the pipeline. Declares execution plans.
- Provides operators to perform ETL (Extract, Transform, and Load) functions.

Apache Pig - Architecture

The language used to analyze data in Hadoop using Pig is known as **Pig Latin**. It is a high-level data processing language that provides a rich set of data types and operators to perform various operations on the data.

To perform a particular task Programmers use Pig, programmers need to write a Pig script using the Pig Latin language and execute them using any of the execution mechanisms (Grunt Shell, UDFs, Embedded). After execution, these scripts will go through a series of transformations applied by the Pig Framework, to produce the desired output.

Internally, Apache Pig converts these scripts into a series of MapReduce jobs, and thus, it makes the programmer's job easy. The architecture of Apache Pig is shown below.



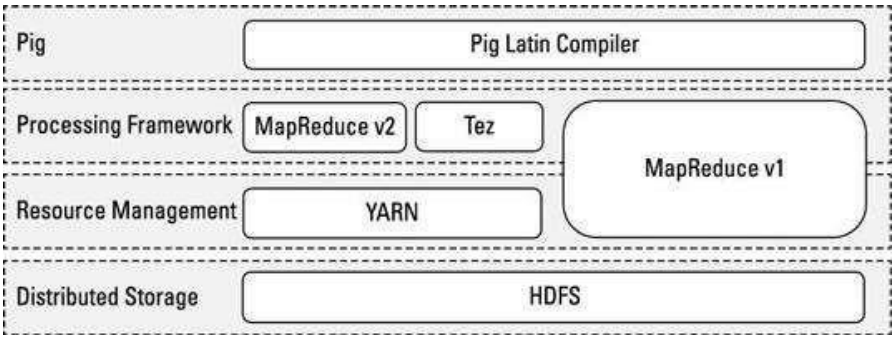
Admiring the pig architecture

A pig is made up of two components:

The language itself: As proof that programmers have a sense of humor, the Programming language for Pig is known as Pig Latin, a high-level language that allows you to write data processing and analysis programs.

The Pig Latin compiler: The Pig Latin compiler converts the Pig Latin code into executable code. The executable code is either in the form of MapReduce jobs or it can spawn a process where a virtual Hadoop instance is created to run the Pig node on a single node.

The sequence of MapReduce programs enables Pig programs to do data processing and analysis in parallel, leveraging Hadoop MapReduce and HDFS. Running the Pig job in the virtual Hadoopinstance is a useful strategy for testing your Pig scripts.



Pig relates to the Hadoop ecosystem

Pig programs can run on MapReduce v1 or MapReduce v2 without any code changes, regardless of what mode your cluster is running. However, Pig scripts can also run using the Tez API instead. Apache Tez provides a more efficient execution framework than MapReduce. YARN enables application frameworks other than MapReduce (like Tez) to run on Hadoop. Hive can also run against the Tez framework.

Apache Pig Components

As shown in the figure, there are various components in the Apache Pig framework. Let us take alook at the major components.

Parser

Initially, the Pig Scripts are handled by the Parser. It checks the syntax of the script, does type checking, and does other miscellaneous checks. The output of the parser will be a DAG Latin statementand logical (directed acyclic graph), which represents the Pig operators.

In the DAG, the logical operators of the script are represented and the data flows are represented as edges.

Optimizer

The logical plan (DAG) is passed to the logical optimizer, which carries out the logical optimizations such as projection and pushdown.

Compiler

The compiler compiles the optimized logical plan into a series of MapReduce jobs.

Execution engine

Finally, the MapReduce jobs are submitted to Hadoop in sorted order. Finally, these MapReducejobs are executed on Hadoop producing the desired results.

Going with the Pig Latin Application Flow

At its core, Pig Latin is a dataflow language, where we define a data stream and a series of transformations that are applied to the data as it flows through your application.

This is in contrast to a control flow language (like C or Java), where we write a series of instructions. In control flow languages, we use constructs like loops and conditional logic (like anif statement). You won't find loops and if statements in Pig Latin. Sample Pig code:

```
A = LOAD 'data_file.txt';

..

B = GROUP ...;

..

C = FILTER ...;

... DUMP B;
..

STORE   INTO 'Results';
```

Looking at each line, in turn, you can see the basic flow of a Pig program.

1. **Load**: we first load (LOAD) the data you want to manipulate. As in a typical MapReduce job, that data is stored in HDFS. For a Pig program to access the data, you first tell Pig what file or files to use. For that task, you use the LOAD 'data_file' command. Here, 'data_file' can specify either an HDFS file or a directory. If a directory is specified, all files in that directory are loaded into theprogram.

If the data is stored in a file format that isn't natively accessible to Pig, you can optionally add theUSING function to the LOAD statement to specify a user-defined function that can read in (and interpret) the data.

2. **Transform**: You run the data through a set of transformations that, way under the hood and far removed from anything you have to concern yourself with, are translated into a set of Map and Reduce tasks. The transformation logic is where all the data manipulation happens. Here, you can FILTER out rows that aren't of interest, JOIN two sets of data files, GROUP data to build aggregations, ORDER results, and do much, much more.

3. **Dump**: Finally, you dump (DUMP) the results to the screen or Store (STORE) the

results in a file somewhere.

Pig Latin Data Model (pig data types)

The data model of Pig Latin is fully nested and it allows complex non-atomic data types such asmaps and tuples. Given below is the diagrammatical representation of Pig Latin's data model.

Atom: An atom is any single value, such as a string or a number — 'Diego', for example. Pig's atomic values are scalar types that appear in most programming languages — int, long, float, double char array, and byte array.

Tuple: A tuple is a record that consists of a sequence of fields. Each field can be of any type — 'Diego', 'Gomez', or 6, for example. Think of a tuple as a row in a table.

Bag: A bag is a collection of non-unique tuples. The schema of the bag is flexible — each tuple inthe collection can contain an arbitrary number of fields, and each field can be of any type.

Map: A map is a collection of key-value pairs. Any type can be stored in the value, and the key needs to be unique. The key of a map must be a char array and the value can be of any type.

Pig Latin operations:

In a Hadoop context, accessing data means allowing developers to load, store, and stream data, whereas transforming data means taking advantage of Pig's ability to group, join, combine, split, filter, and sort data.

Data access:

Operator	Explanation
LOAD/STORE	Read and write data to file system
DUMP	Write output to standard output (stdout)
STREAM	Send all records through an external binary
FOREACH	Apply an expression to each record and outputone or more records

FILTER	Apply predicate and remove records that don'tmeet the condition
GROUP/COGROUP	Aggregate records with the same key from oneor more inputs
JOIN	Join two or more records based on a condition

Transformations

Operator	Explanation
CROSS	Cartesian product of two or more inputs
ORDER	Sort records based on key
DISTINCT	Remove duplicate records
UNION	Merge two data sets
SPLIT	Divide data into two or more bags based onpredicate
LIMIT	Subset the number of records

Operators for debugging and troubleshooting

Operator	Explanation
DESCRIBE	Return the schema of a relation
DUMP	Dump the contents of a relation to the screen
EXPLAIN	Display the MapReduce execution plans.

Apache Pig Execution Modes

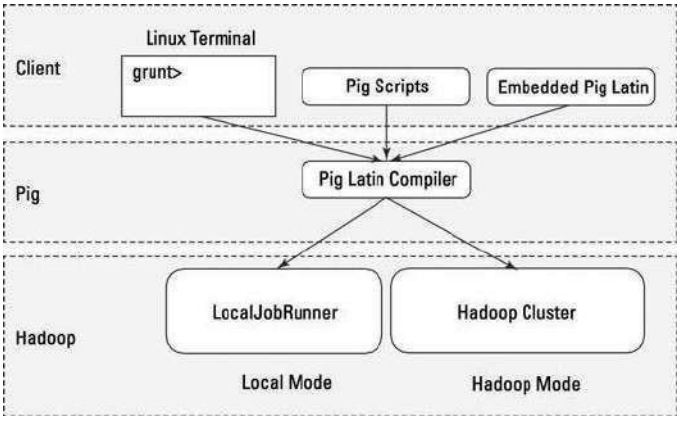
We can run Apache Pig in two modes, namely, Local Mode and HDFS mode.

Local Mode

In this mode, all the files are installed and run from your localhost and local file system. There is no need for Hadoop or HDFS. This mode is generally used for testing purposes.

MapReduce Mode

MapReduce mode is where we load or process the data that exists in the Hadoop File System (HDFS) using Apache Pig. In this mode, whenever we execute the Pig Latin statements to process the data, a MapReduce job is invoked in the back-end to perform a particular operation on the datathat exists in the HDFS.



Apache Pig Execution Mechanisms

Apache Pig scripts can be executed in three ways, namely, interactive mode, batch mode, and embedded mode.

- **Interactive Mode** (Grunt shell) − You can run Apache Pig in interactive mode using the Grunt shell. In this shell, you can enter the Pig Latin statements and get the output (using the Dump operator).
- **Batch Mode** (Script) − You can run Apache Pig in Batch mode by writing the Pig Latin script in a single file with the .pig extension.
- **Embedded Mode** (UDF) − Apache Pig provides the provision of defining our functions (**U**ser **D**efined **F**unctions) in programming languages such as Java, and using them in our script.