**SESSION 7: APACHE PIG**

**ASSIGNMENT 1**

1. **Why Map-reduce programming is needed in PIG programming?**

Pig is an application that works on top of Mapreduce, YARN or TEZ. It is an abstraction over Mapreduce, i.e., it hides the underlying complexities from the end user. Pig Latin is Pig's language that allows developers to sort, join, parse, transform and calculate unstructured and semi-structured data in MapReduce all while using a language similar to SQL versus Java.

Pig, can be thought of as a compiler, which takes PIG latin scripts and transforms them into JAVA (Mapreduce).

1. **What are advantages of Pig over Mapreduce?**

* Decrease in development time
* Easy to learn
* Less complex – Joins and Ordering comprise 8-9 lines of code in PIG latin and take a few minutes to write and debug. In mapreduce, the same code will span hundreds of lines of code and take hours to develop.
* Lazy evaluation – Unless you don’t produce any output file or output any message, it doesn’t get evaluated. This leaves better scope for PIG optimizer.
* Developers not well versed with JAVA can easily work on PIG.

1. **What is PIG engine and what is its importance?**

PIG engine is one of the major components of PIG. It acts as an interpreter between PIG latin script and mapreduce jobs. It accepts the PIG latin script as input and converts those scripts into MapReduce jobs.

1. **What are the modes of PIG execution?**

Apache PIG has two modes of execution, namely –

* Local Mode

In this mode, the entire PIG job runs in a single JVM picking the local UNIX path for execution. All the files are installed and run from the Local File System. There is no need of 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 data that exists in the HDFS.

While running PIG in mapreduce mode, one must make sure the job history server is running. It helps us to view previous MR job details.

1. **What is Grunt shell in Pig?**

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

Grunt is PIG’s interactive shell. It enables the users to enter PIG latin interactively. It also provides a shell for PIG users to interact with the HDFS.

Grunt shell can be invoked by **pig/pig -x mapreduce** for mapreduce mode and **pig -x local** for the local mode.

1. **What are the features of Pig Latin language?**

* PIG Latin is a dataflow language
* PIG is self-optimizing

Because the system automatically optimizes execution of Pig jobs, the user can focus on semantics.

* It comes with a rich set of operators to perform functions like join, sort, filter, etc.
* Extensible – PIG users can create custom functions to meet their particular processing requirements.
* PIG latin can analyze both structured, as well as unstructured data.
* UDFs – PIG provides the facility to create User-Defined Functions in other programming languages such as JAVA and invoke or embed them in PIG scripts.

1. **Are Pig latin commands case sensitive?**

PIG latin is neither completely case sensitive, nor completely insensitive. Keywords in PIG latin are not case sensitive. For example, LOAD is equivalent to load.

But relation and field names are case sensitive.

So, A = load ‘abc.txt’ is not equivalent to a = load ‘abc.txt’;

1. **What is a data flow language?**

Dataflow programming is a programming paradigm that models a program as a directed graph of the data flowing between operations, thus implement dataflow principles and architecture.

Dataflow programming emphasizes the movement of data and models programs as a series of connections. Explicitly defined inputs and outputs connect operations. An operation runs as soon as all of its inputs become valid. Thus, dataflow languages are inherently parallel and can work well in large, decentralized systems.