Assignment 7.1

1. Why Map-reduce program is needed in Pig Programming?

What does a map-reduce program do

* The system takes input from a file system and splits it up across separate Map nodes.
* The Map-code runs and generates an output for each Map node.
* The output is a set of <key,value> pairs which form input to reduce code
* The reduce code is run and generates an output for each reduce node.
* The system take output from each reduce node, aggregates and produces the results.

So basically there is enough programming involved here. We could use native Hadoop libraries and run the Map-Reduce program written in Java and have the results, we could also use Pig, Hive as well in many a cases.

Pig is a high-level procedural language. The programming language is referred to as Pig Latin. This provides and abstraction for the MapReduce programming. It has a style very similar to querying used in SQL.

Pig simplifies the use of Hadoop by allowing SQl-like syntaxes.

It allows parallel processing

It has several operators for usual data operations like – JOIN, SORT, FILTER etc.

It uses the HDFS file system.

Data from one or more inputs is read and processed.

Pig uses map-reduce internally to process the data.

How PIG works and where map-reduce comes in

First comes the parser, which checks for syntax of the script/code written, the output of this is all PigLatin statements and operators. Next, the optimizer, where the logical optimizations are done. Then the complier, which complies the optimized logical plan into a series of MapReduce jobs. The MapReduce jobs are then executed on Hadoop producing desired results. Here it accesses the HDFS file system.

What Pig does is create a layer of abstraction from the complex MapReduce program which would have to written to solve the same case. It

1. Reduces the complexity
2. Simple SQL like statement
3. Reduced the code length to write and debug largely thus saving developer time.

Thus Pig needs the MapReduce program since it is working with HDFS internally. Pig is a high-level platform that makes many Hadoop data analysis issues easier to execute. A program written in Pig Latin is a data flow language, which need to execute the query. So, when a program is written in Pig Latin, Pig compiler converts the program into MapReduce jobs.

1. What are advantages of pig over MapReduce?

Advantages of Pig over MapReduce

1. Pig is a high level data flow program whereas MapReduce is low-level data processing
2. Higher level of abstraction – easy to learn, use, read, debug
3. Pig provides simpler programing methods unlike MapReduce which uses complex Java implementations and logic
4. Reduces the lines of code to at least 1/20th in Pig when compared to native
5. Pig provides nested data types like tuples, bags, and maps that are missing from MapReduce.
6. Pig provides many built-in operators to support data operations like joins, filters, ordering, sorting etc. Whereas to perform the same function in MapReduce is a humongous task.
7. Less worry about version mismatch etc (have multiple client libraries installed)

3. What is pig engine and what is its importance?

Once the code is through the parser, optimizer and then complier, then comes the Pig engine. Here the MapReduce jobs are submitted to Hadoop. Its sorted. Then the results of the MapReduce is produced.

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 can perform 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 own functions for reading, writing, and processing data.

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

4. What are the modes of Pig execution?

Execution modes

1. MapReduce/Hadoop mode

Pig jobs run as MapReduce jobs taking inputs from HDFS. Data gets loaded from the HDFS, MapReduce program gets run at the backend.

pig –x mapreduce

is used to run Pig in MapReduce mode.

1. local mode

here the Pig job is run as a single JVM using the input file from local path. Data gets loaded from local file system and no mapreduce program. This is faster execution, good for testing purposes.

Use pig OR pig –x local to run Pig in local mode.

5. What is grunt shell in Pig?

* an interactive shell for executing Pig commands. After invoking the Grunt shell, you can run your Pig scripts in the shell.
* mainly used to write Pig Latin scripts
* many utility commands like clear, help, history, quit, and set, exec, kill, and run  are available
* eg grunt>my\_rel = LOAD ‘/home/acadgild/pig/employee\_details.txt’ USING PigStorage(,) AS (emp\_id:int, emp\_name:chararray, emp\_salary:int, emp\_rating:int)

1. What are the features of Pig Latin language?

Pig Latin language

* Language for expressing data flows
* It is a command based scripting language and designed specifically for data transformation and flow
* Has simple data types like – int, long, float, double, chararray, bytearray
* Has complex data types – tuple, bag, map
* Single value is called – atom
* Tuple – group of fields, like a records
* Bag – unordered set of tuples
* Map – set of <key,value> pairs
* Relation – outer bag of tuples
* Has operators for data processing operations – load, dump, filter, group etc
* Load – loads data from file system

1. Is Pig latin commands case sensitive?

No, Pig Latin commands are not case-sensitive.

A = LOAD ‘data.file’

A = load ‘data.file’

Are the same.

What is a data flow language?

The definition of data flow language as given by Wikipedia is

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

What this means is looking at what control language does :

In a control flow language, you have a stream of instructions which operate on external data. Conditional execution, jumps and procedure calls change the instruction stream to be executed. This could be seen as instructions flowing through data (eg instructions operate on registers which are loaded with data by instructions - the data is static unless the instruction stream moves it). A control flow "if" statement jumps to the correct branch in the instruction stream, but the data does not get moved.

In a dataflow language, you have a stream of data which is passed from instruction to instruction to be processed. Conditional execution, jumps and procedure calls route the data to different instructions. This could be seen as data flowing through otherwise static instructions like how electrical signals flow through circuits or water flows through pipes. A dataflow "if" statement would route the data to the correct branch.

Example : Unix pipes uses this.