**Pig Questions & Answers**

**Important Points:**  
Apache Pig is a platform for analyzing large data sets that consists of a high-level language for expressing data analysis programs, coupled with infrastructure for evaluating these programs.

The salient property of Pig programs is that their structure is amenable to substantial parallelization, which in turns enables them to handle very large data sets.   
  
**1)What is Pig?**   
**Answer:** Apache Pig is a platform, used to analyze large data sets representing them as data flows. It is designed to provide an abstraction over MapReduce, reducing the complexities of writing a MapReduce task using Java programming. We can perform data manipulation operations very easily in Hadoop using Apache Pig. Apache Pig has two main components – the Pig Latin language and the Pig Run-time Environment, in which Pig Latin programs are executed.   
  
**2)How can I pass a specific hadoop configuration parameter to Pig?**   
**Answer:** There are multiple places you can pass hadoop configuration parameter to Pig. Here is a list from high priority to low priority (configuration in high priority will override the configuration in low priority):  
1. set command  
2. -P properties\_file  
3. pig.properties  
4. java system property/environmental variable  
5. Hadoop configuration file: hadoop-site.xml/core-site.xml/hdfs-site.xml/mapred-site.xml, or Pig specific hadoop configuration file: pig-cluster-hadoop-site.xml  
  
**3)I already register my LoadFunc/StoreFunc jars in "register" statement, but why I still get "Class Not Found" exception?**   
**Answer:** Try to put your jars in PIG\_CLASSPATH as well. "register" guarantees your jar will be shipped to backend. But in the frontend, you still need to put the jars in CLASSPATH by setting "PIG\_CLASSPATH" environment variable.   
  
**4)How can I load data using Unicode control characters as delimiters?**   
**Answer:** The first parameter to PigStorage is the dataset name, the second is a regular expression to describe the delimiter. We used `String.split(regex, -1)` to extract fields from lines. See java.util.regex.Pattern for more information on the way to use special characters in regex.  
If you are loading a file which contains Ctrl+A as separators, you can specify this to PigStorage using the Unicode notation.  
LOAD 'input.dat' USING PigStorage('\u0001')as (x,y,z);  
**5)How do I control the number of mappers?**   
**Answer:** It is determined by your InputFormat. If you are using PigStorage, FileInputFormat will allocate at least 1 mapper for each file. If the file is large, FileInputFormat will split the file into smaller trunks. You can control this process by two hadoop setting: "mapred.min.split.size", "mapred.max.split.size". In addition, after InputFormat tells Pig all the splits information, Pig will try to combine small input splits into one mapper. This process can be controlled by "pig.noSplitCombination" and "pig.maxCombinedSplitSize".   
  
**6)How do I make my Pig jobs run on a specified number of reducers?**   
**Answer:** You can achieve this with the PARALLEL clause.  
For example: C = JOIN A by url, B by url PARALLEL 50.  
Besides PARALLEL clause, you can also use "set default\_parallel" statement in Pig script, or set "mapred.reduce.tasks" system property to specify default parallel to use. If none of these values are set, Pig will only use 1 reducers. (In Pig 0.8, we change the default reducer from 1 to a number calculated by a simple heuristic for foolproof purpose)  
  
**7)Can I do a numerical comparison while filtering?**   
**Answer:** Yes, you can choose between numerical and string comparison. For numerical comparison use the operators =, and for string comparisons use eq, neq etc.   
  
**8)Does Pig support regular expressions?**   
**Answer:** Pig does support regular expression matching via the `matches` keyword. It uses java.util.regex matches which means your pattern has to match the entire string (e.g. if your string is `"hi fred"` and you want to find `"fred"` you have to give a pattern of `".\*fred"` not `"fred"`).   
  
**9)How do I prevent failure if some records don't have the needed number of columns?**   
**Answer:** You can filter away those records by including the following in your Pig program:  
A = LOAD 'foo' USING PigStorage('\t');  
B = FILTER A BY ARITY(\*) < 5;  
This code would drop all records that have fewer than five (5) columns.  
  
**10)Is there any difference between `==` and `eq` for numeric comparisons?**   
**Answer:** There is no difference when using integers. However, `11.0` and `11` will be equal with `==` but not with `eq`. 

**11)Is there an easy way for me to figure out how many rows exist in a dataset from it's alias?**   
**Answer:** You can run the following set of commands, which are equivalent to `SELECT COUNT(\*)` in SQL:  
a = LOAD 'mytestfile.txt';  
b = GROUP a ALL;  
c = FOREACH b GENERATE COUNT(a.$0);  
  
**12)Does Pig allow grouping on expressions?**   
**Answer:** Pig allows grouping of expressions. For example:  
grunt> a = LOAD 'mytestfile.txt' AS (x,y,z);  
grunt> DUMP a;  
(1,2,3)  
(4,2,1)  
(4,3,4)  
(4,3,4)  
(7,2,5)  
(8,4,3)  
  
b = GROUP a BY (x+y);  
(3.0,{(1,2,3)})  
(6.0,{(4,2,1)})  
(7.0,{(4,3,4),(4,3,4)})  
(9.0,{(7,2,5)})  
(12.0,{(8,4,3)})  
If the grouping is based on constants, the result is the same as GROUP ALL except the group-id is replaced by the constant.  
grunt> b = GROUP a BY 4;  
(4,{(1,2,3),(4,2,1),(4,3,4),(4,3,4),(7,2,5),(8,4,3)})  
  
**13)Is there a way to check if a map is empty?**  
**Answer:** In Pig 2.0 you can test the existence of values in a map using the null construct:  
m#'key' is not null  
  
**14)I load data from a directory which contains different file. How do I find out where the data comes from?**   
**Answer:** You can write a LoadFunc which append filename into the tuple you load.  
Eg,  
A = load '\*.txt' using PigStorageWithInputPath();  
Here is the LoadFunc:  
public class PigStorageWithInputPath extends PigStorage {  
Path path = null;  
  
  
@Override  
public void prepareToRead(RecordReader reader, PigSplit split) {  
super.prepareToRead(reader, split);  
path = ((FileSplit)split.getWrappedSplit()).getPath();  
}  
  
@Override  
public Tuple getNext() throws IOException {  
Tuple myTuple = super.getNext();  
if (myTuple != null)  
myTuple.append(path.toString());  
return myTuple;  
}  
}  
  
**15)How can I calculate a percentage (partial aggregate / total aggregate)?**   
**Answer:** The challenge here is to get the total aggregate into the same statement as the partial aggregate. The key is to cast the relation for the total aggregate to a scalar:  
A = LOAD 'sample.txt' AS (x:int, y:int);  
B = foreach (group A all) generate COUNT(A) as total;  
C = foreach (group A by x) generate group as x, (double)COUNT(A) / (double) B.total as percentage;  
  
**16)How can I pass a parameter with space to a pig script?**   
**Answer:** # Following should work  
-p \"NAME='Firstname Lastname'\"  
-p \"NAME=Firstname\ Lastname\"  
  
  
**17)What is the difference between logical and physical plans?**   
**Answer:** Pig undergoes some steps when a Pig Latin Script is converted into MapReduce jobs by the compiler. Logical and Physical plans are created during the execution of a pig script.  
After performing the basic parsing and semantic checking, the parser produces a logical plan and no data processing takes place during the creation of a logical plan. The logical plan describes the logical operators that have to be executed by Pig during execution. For each line in the Pig script, syntax check is performed for operators and a logical plan is created. If an error is encountered, an exception is thrown and the program execution ends.  
A logical plan contains a collection of operators in the script, but does not contain the edges between the operators.  
After the logical plan is generated, the script execution moves to the physical plan where there is a description about the physical operators, Apache Pig will use, to execute the Pig script. A physical plan is like a series of MapReduce jobs, but the physical plan does not have any reference on how it will be executed in MapReduce.  
  
**18)How Pig programming gets converted into MapReduce jobs?**   
**Answer:** 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 an execution engine to execute the query. So, when a program is written in Pig Latin, Pig compiler converts the program into MapReduce jobs.   
  
**19)What are the components of Pig Execution Environment?**   
**Answer:** The components of Apache Pig Execution Environment are:   
Pig Scripts: Pig scripts are submitted to the Apache Pig execution environment which can be written in Pig Latin using built-in operators and UDFs can be embedded in it.  
Parser: The Parser does the type checking and checks the syntax of the script. The parser outputs a DAG (directed acyclic graph). DAG represents the Pig Latin statements and logical operators.  
Optimizer: The Optimizer performs the optimization activities like split, merge, transform, reorder operators, etc. The optimizer provides the automatic optimization feature to Apache Pig. The optimizer basically aims to reduce the amount of data in the pipeline.  
Compiler: The Apache Pig compiler converts the optimized code into MapReduce jobs automatically.  
Execution Engine: Finally, the MapReduce jobs are submitted to the execution engine. Then, the MapReduce jobs are executed and the required result is produced.  
  
**20)What are the different ways of executing Pig script?**   
**Answer:** There are three ways to execute the Pig script:  
Grunt Shell: This is Pig’s interactive shell provided to execute all Pig Scripts.  
Script File: Write all the Pig commands in a script file and execute the Pig script file. This is executed by the Pig Server.  
Embedded Script: If some functions are unavailable in built-in operators, we can programmatically create User Defined Functions (UDF) to bring that functionality using other languages like Java, Python, Ruby, etc. and embed it in the Pig Latin Script file. Then, execute that script file.  
  
**21)What are the data types of Pig Latin?**   
**Answer:** Pig Latin can handle both atomic data types like int, float, long, double etc. and complex data types like tuple, bag and map.  
Atomic or scalar data types are the basic data types which are used in all the languages like string, int, float, long, double, char[], byte[]. These are also called the primitive data types.  
The complex data types supported by Pig Latin are:  
Tuple: Tuple is an ordered set of fields which may contain different data types for each field.  
Bag: A bag is a collection of a set of tuples and these tuples are a subset of rows or entire rows of a table.  
Map: A map is key-value pairs used to represent data elements. The key must be a chararray [] and should be unique like column name, so it can be indexed and value associated with it can be accessed on the basis of the keys. The value can be of any data type.  
  
**22)Is it possible to pivot a table in one pass in Apache Pig.  
Input:  
Id Column1 Column2 Column3  
1 Row11 Row12 Row13  
2 Row21 Row22 Row23  
Output:  
Id Name Value  
1 Column1 Row11  
1 Column2 Row12  
1 Column3 Row13  
2 Column1 Row21  
2 Column2 Row22  
2 Column3 Row23  
Answer:** You can do it in 2 ways: 1. Write a UDF which returns a bag of tuples. It will be the most flexible solution, but requires Java code; 2. Write a rigid script like this:  
inpt = load '/pig\_fun/input/pivot.txt' as (Id, Column1, Column2, Column3);  
bagged = foreach inpt generate Id, TOBAG(TOTUPLE('Column1', Column1), TOTUPLE('Column2', Column2), TOTUPLE('Column3', Column3)) as toPivot;  
pivoted\_1 = foreach bagged generate Id, FLATTEN(toPivot) as t\_value;  
pivoted = foreach pivoted\_1 generate Id, FLATTEN(t\_value);  
dump pivoted;  
Running this script got me following results:  
(1,Column1,11)  
(1,Column2,12)  
(1,Column3,13)  
(2,Column1,21)  
(2,Column2,22)  
(2,Column3,23)  
(3,Column1,31)  
(3,Column2,32)  
(3,Column3,33)  
  
  
**23)How to Load multiple files from a date range (part of the directory structure)I have the following scenario-  
Sample HDFS directory structure:  
/user/training/test/20100810/data files  
/user/training/test/20100811/data files  
/user/training/test/20100812/data files  
/user/training/test/20100813/data files  
/user/training/test/20100814/data files  
As you can see in the paths listed above, one of the directory names is a date stamp.  
Problem: I want to load files from a date range say from 20100810 to 20100813.  
Answer:** The path expansion is done by the shell. One common way to solve your problem is to simply use Pig parameters (which is a good way to make your script more resuable anyway):  
shell:  
pig -f script.pig -param input=/user/training/test/{20100810..20100812}  
script.pig:  
temp = LOAD '$input' USING SomeLoader() AS ();  
  
**24)how to reference columns in a FOREACH after a JOIN?  
A = load 'a.txt' as (id, a1);  
B = load 'b.txt as (id, b1);  
C = join A by id, B by id;  
D = foreach C generate id,a1,b1;  
dump D;  
4th line fails on: Invalid field projection. Projected field [id] does not exist in schema, How to fix this?  
Answer:** Solution:  
A = load 'a.txt' as (id, a1);  
B = load 'b.txt as (id, b1);  
C = join A by id, B by id;  
D = foreach C generate A::id,a1,b1;  
dump D;  
  
**25)how to include external jar file using PIG**   
**Answer:** register /local/path/to/myJar.jar 

**26)Removing duplicates using PigLatin   
Input:  
User1 8 NYC  
User1 9 NYC  
User1 7 LA  
User2 4 NYC  
User2 3 DC  
Output:  
User1 8 NYC  
User2 4 NYC  
Answer:** In order to select one record per user (any record) you could use a GROUP BY and a nested FOREACH with LIMIT. Ex:  
  
inpt = load '......' ......;  
user\_grp = GROUP inpt BY $0;  
filtered = FOREACH user\_grp {  
top\_rec = LIMIT inpt 1;  
GENERATE FLATTEN(top\_rec);  
};  
  
**27)Currently, when we STORE into HDFS, it creates many part files.Is there any way to store out to a single CSV file in Apache Pig?**   
**Answer:** You can do this in a few ways:  
To set the number of reducers for all Pig opeations, you can use the default\_parallel property - but this means every single step will use a single reducer, decreasing throughput:  
set default\_parallel 1;  
Prior to calling STORE, if one of the operations execute is (COGROUP, CROSS, DISTINCT, GROUP, JOIN (inner), JOIN (outer), and ORDER BY), then you can use the PARALLEL 1 keyword to denote the use of a single reducer to complete that command:  
GROUP a BY grp PARALLEL 1;

**28)I have data that's already grouped and aggregated, it looks like so:  
user value count  
---- -------- ------  
Alice third 5  
Alice first 11  
Alice second 10  
Alice fourth 2  
Bob second 20  
Bob third 18  
Bob first 21  
Bob fourth 8  
  
For every user (Alice and Bob), I want retrieve their top n values (let's say 2), sorted terms of 'count'. So the desired output I want is this:  
  
Alice first 11  
Alice second 10  
Bob first 21  
Bob second 20  
How can I accomplish this in Apache Pig?**   
**Answer:** One approach is  
records = LOAD '/user/nubes/ncdc/micro-tab/top.txt' AS (user:chararray,value:chararray,counter:int);  
grpd = GROUP records BY user;  
top3 = foreach grpd {  
sorted = order records by counter desc;  
top = limit sorted 2;  
generate group, flatten(top);  
};  
  
Input is:  
Alice third 5  
Alice first 11  
Alice second 10  
Alice fourth 2  
Bob second 20  
Bob third 18  
Bob first 21  
Bob fourth 8  
Output is:  
(Alice,Alice,first,11)  
(Alice,Alice,second,10)  
(Bob,Bob,first,21)  
(Bob,Bob,second,20)  
  
**29)Find if a string is present inside another string in Pig**   
**Answer:** You can use this :  
X = FILTER A BY (f1 matches '.\*the\_word\_you're\_looking\_for.\*');   
  
**30)how to do Transpose in corresponding few columns in pig?  
Input:  
id jan feb march  
1 j1 f1 m1  
2 j2 f2 m2  
3 j3 f3 m3  
Output:  
id value month  
1 j1 jan  
1 f1 feb  
1 m1 march  
2 j2 jan  
2 f2 feb  
2 m2 march  
3 j3 jan  
3 f3 feb  
3 m3 march  
Answer:** PigScript:  
A = LOAD 'input.txt' USING PigStorage() AS (id,month1,month2,month3);  
B = FOREACH A GENERATE FLATTEN(TOBAG(TOTUPLE(id,month1,'jan'),TOTUPLE(id,month2,'feb'),TOTUPLE(id,month3,'mar')));  
DUMP B;  
Output:  
(1,j1,jan)  
(1,f1,feb)  
(1,m1,mar)  
(2,j2,jan)  
(2,f2,feb)  
(2,m2,mar)  
(3,j3,jan)  
(3,f3,feb)  
(3,m3,mar)  
  
  
**31)What is the difference between Store and dump commands?**   
**Answer:** Dump command after process the data displayed on the terminal, but it’s not stored anywhere. Where as Store stored in local file system or HDFS and output execute in a folder. In the protection environment most often hadoop developer used ‘store’ command to store data in in the HDFS.   
  
**32)How to debug a pig script?**   
**Answer:** There are several method to debug a pig script. Simple method is step by step execution of a relation and then verify the result. These commands are useful to debug a pig script.  
DUMP - Use the DUMP operator to run (execute) Pig Latin statements and display the results to your screen.  
ILLUSTRATE - Use the ILLUSTRATE operator to review how data is transformed through a sequence of Pig Latin statements. ILLUSTRATE allows you to test your programs on small datasets and get faster turnaround times.  
EXPLAIN - Use the EXPLAIN operator to review the logical, physical, and map reduce execution plans that are used to compute the specified relationship.  
DESCRIBE - Use the DESCRIBE operator to view the schema of a relation. You can view outer relations as well as relations defined in a nested FOREACH statement.  
  
**33)What are the limitations of the Pig?**   
**Answer:** Limitations of the Apache Pig are:  
As the Pig platform is designed for ETL-type use cases, it’s not a better choice for real-time scenarios.  
Apache Pig is not a good choice for pinpointing a single record in huge data sets.  
Apache Pig is built on top of MapReduce, which is batch processing oriented.  
  
**34)What is BloomMapFile used for?**   
**Answer:** The BloomMapFile is a class that extends MapFile. So its functionality is similar to MapFile. BloomMapFile uses dynamic Bloom filters to provide quick membership test for the keys. It is used in Hbase table format.   
  
**35)What is the difference between GROUP and COGROUP operators in Pig?**   
**Answer:** Group and Cogroup operators are identical. For readability, GROUP is used in statements involving one relation and COGROUP is used in statements involving two or more relations. Group operator collects all records with the same key. Cogroup is a combination of group and join, it is a generalization of a group instead of collecting records of one input depends on a key, it collects records of n inputs based on a key. At a time, we can Cogroup up to 127 relations. 

**36)Give some list of relational operators used in Pig?**   
**Answer:** COGROUP: Joins two or more tables and then perform GROUP operation on the joined table result.  
CROSS: CROSS operator is used to compute the cross product (Cartesian product) of two or more relations.  
DISTINCT: Removes duplicate tuples in a relation.  
FILTER: Select a set of tuples from a relation based on a condition.  
FOREACH: Iterate the tuples of a relation, generating a data transformation.  
GROUP: Group the data in one or more relations.  
JOIN: Join two or more relations (inner or outer join).  
LIMIT: Limit the number of output tuples.  
LOAD: Load data from the file system.  
ORDER: Sort a relation based on one or more fields.  
SPLIT: Partition a relation into two or more relations.  
STORE: Store data in the file system.  
UNION: Merge the content of two relations. To perform a UNION operation on two relations, their columns and domains must be identical.  
  
**37)Can we process vast amount of data in local mode? Why?**   
**Answer:** No, System has limited fixed amount of storage, where as Hadoop can handle vast amount of data. So Pig -x Mapreduce mode is the best choice to process vast amount of data.   
  
**38)Explain about the different complex data types in Pig?**   
**Answer:** Apache Pig supports 3 complex data types-  
Maps- These are key, value stores joined together using #.  
Tuples- Just similar to the row in a table, where different items are separated by a comma. Tuples can have multiple attributes.  
Bags- Unordered collection of tuples. Bag allows multiple duplicate tuples.  
  
**39)Differentiate between the logical and physical plan of an Apache Pig script?**   
**Answer:** Logical and Physical plans are created during the execution of a pig script. Pig scripts are based on interpreter checking. Logical plan is produced after semantic checking and basic parsing and no data processing takes place during the creation of a logical plan. For each line in the Pig script, syntax check is performed for operators and a logical plan is created. Whenever an error is encountered within the script, an exception is thrown and the program execution ends, else for each statement in the script has its own logical plan.  
A logical plan contains collection of operators in the script but does not contain the edges between the operators.  
After the logical plan is generated, the script execution moves to the physical plan where there is a description about the physical operators, Apache Pig will use, to execute the Pig script. A physical plan is more or less like a series of MapReduce jobs but then the plan does not have any reference on how it will be executed in MapReduce. During the creation of physical plan, cogroup logical operator is converted into 3 physical operators namely –Local Rearrange, Global Rearrange and Package. Load and store functions usually get resolved in the physical plan.  
  
**40)What do you understand by an inner bag and outer bag in Pig?**   
**Answer:** A relation inside a bag is referred to as inner bag and outer bag is just a relation in Pig   
  
**41)Explain the difference between COUNT\_STAR and COUNT functions in Apache Pig?**   
**Answer:** COUNT function does not include the NULL value when counting the number of elements in a bag, whereas COUNT\_STAR (0 function includes NULL values while counting.   
  
**42)Explain about the scalar datatypes in Apache Pig.**   
**Answer:** integer, float, double, long, bytearray and char array are the available scalar datatypes in Apache Pig.   
  
**43)Is it posible to join multiple fields in pig scripts?**   
**Answer:** Yes,Join select records from one input and join with another input.This is done by indicating keys for each input. When those keys are equal, the two rows are joined.  
input2 = load ‘daily’ as (exchanges, stocks);  
input3 = load ‘week’ as (exchanges, stocks);  
grpds = join input2 by stocks,input3 by stocks;  
  
we can also join multiple keys  
example:  
input2 = load ‘daily’ as (exchanges, stocks);  
input3 = load ‘week’ as (exchanges, stocks);  
grpds = join input2 by (exchanges,stocks),input3 by (exchanges,stocks);  
  
**44)What are the different String functions available in pig?**   
**Answer:** Below are most commonly used STRING pig functions  
UPPER  
LOWER  
TRIM  
SUBSTRING  
INDEXOF  
STRSPLIT  
LAST\_INDEX\_OF  
  
  
**45)While writing evaluate UDF, which method has to be overridden?**   
**Answer:** While writing UDF in pig, you have to override the method exec() and the base class can be different, while writing filter UDF, you will have to extend FilterFunc and for evaluate UDF, you will have to extend the EvalFunc.EvaluFunc is parameterized and must provide the return type also.   
  
**46)What is a skewed join?**   
**Answer:** Whenever you want to perform a join with a skewed dataset i.e., a particular value will be repeated many times.  
Suppose, if you have two datasets which contains the details about city and the person living in that city. The second dataset contains the details of city and the country.  
So automatically city name will be repeated multiple times based on the population of the city and if you want to perform join using the city column then a particular reducer will receive a lot of values for that particular city.  
In the skewed dataset, the left input on the join predicate will be divided and even if you have skeweness in the data your data will be split across different machines and the input on the right side will be duplicated and split across different machines and in this way skewed join is handled in the Pig.  
  
**47)Write a word count program in pig?**  
**Answer:** lines = LOAD ‘/user/hadoop/HDFS\_File.txt’ AS (line:chararray);  
words = FOREACH lines GENERATE FLATTEN(TOKENIZE (line)) as word;  
grouped = GROUP words by word;  
wordcount = FOREACH grouped GENERATE group, COUNT (words);  
DUMP wordcount;  
  
**48)What is the difference between Pig Latin and HiveQL ?**  
**Answer:** Pig Latin:  
Pig Latin is a Procedural language  
Nested relational data model  
Schema is optional  
HiveQL:  
HiveQL is Declarative  
HiveQL flat relational  
Schema is required  
  
**49)Does Pig support multi-line commands?**  
**Answer:** Yes, pig supports both single line and multi-line commands. In single line command it executes the data, but it doesn’t store in the file system, but in multiple lines commands it stores the data into ‘/output’;/\* , so it can store the data in HDFS. 

**50)What is the function of UNION and SPLIT operators? Give examples.**   
**Answer:** Union operator helps to merge the contents of two or more relations.  
Syntax: grunt> Relation\_name3 = UNION Relation\_name1, Relation\_name2  
Example: grunt> INTELLIPAAT = UNION intellipaat\_data1.txt intellipaat\_data2.txt  
SPLIT operator helps to divide the contents of two or more relations.  
Syntax: grunt> SPLIT Relationa1\_name INTO Relationa2\_name IF (condition1), Relation2\_name (condition2);  
Example: SPLIT student\_details into student\_details1 if marks <35, student\_details2 if (8590);