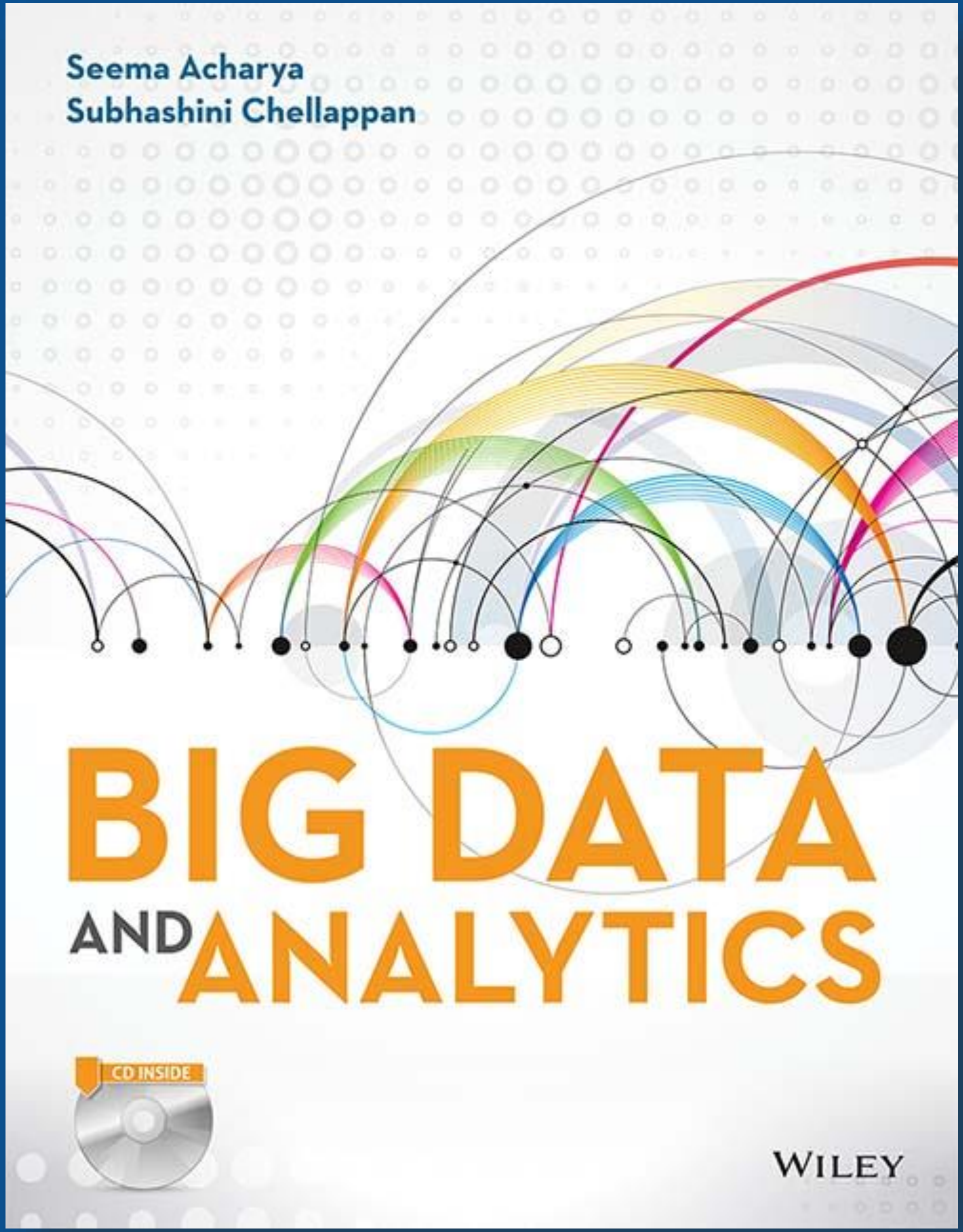


Big Data And Analytics

Seema Acharya
Subhashini Chellappan

Big Data and Analytics by Seema Acharya and Subhashini Chellappan

Chapter 10

Introduction to Pig

Learning Objectives and Learning Outcomes

Learning Objectives	Learning Outcomes
Introduction to Pig 1. To study the key features and anatomy of Pig. 2. To study the execution modes of Pig. 3. To study the various relational operators in pig.	a) To have an easy comprehension on when to use and when NOT to use Pig. b) To be able to differentiate between Pig and Hive.

Session Plan

Lecture time	90 to 120 minutes
Q/A	15 minutes

Agenda

- What is Pig?
    - Key Features of Pig
- The Anatomy of Pig
- Pig on Hadoop
- Pig Philosophy
- Pig Latin Overview
    - Pig Latin Statements
    - Pig Latin: Identifiers
    - Pig Latin: Comments
- Data Types in Pig
    - Simple Data Types
    - Complex Data Types

Agenda

- Running Pig
- Execution Modes of Pig
- Relational Operators
- Eval Function
- Piggy Bank
- When to use Pig?
- When NOT to use Pig?
- Pig versus Hive

What is Pig?

What is Pig?

Apache Pig is a platform for data analysis.

It is an alternative to Map Reduce Programming.

Features of Pig

Features of Pig

- It provides an **engine** for executing **data flows** (how your data should flow). Pig processes data in parallel on the Hadoop cluster.

- It provides a language called “**Pig Latin**” to express data flows.

- Pig Latin contains operators for many of the traditional data operations such as join, filter, sort, etc.

- It allows users to develop their own functions (User Defined Functions) for reading, processing, and writing data.

The Anatomy of Pig

The Anatomy of Pig

The main components of Pig are as follows:

- Data flow language (**Pig Latin**).
- Interactive shell where you can type Pig Latin statements (**Grunt**).
- Pig interpreter and execution engine.

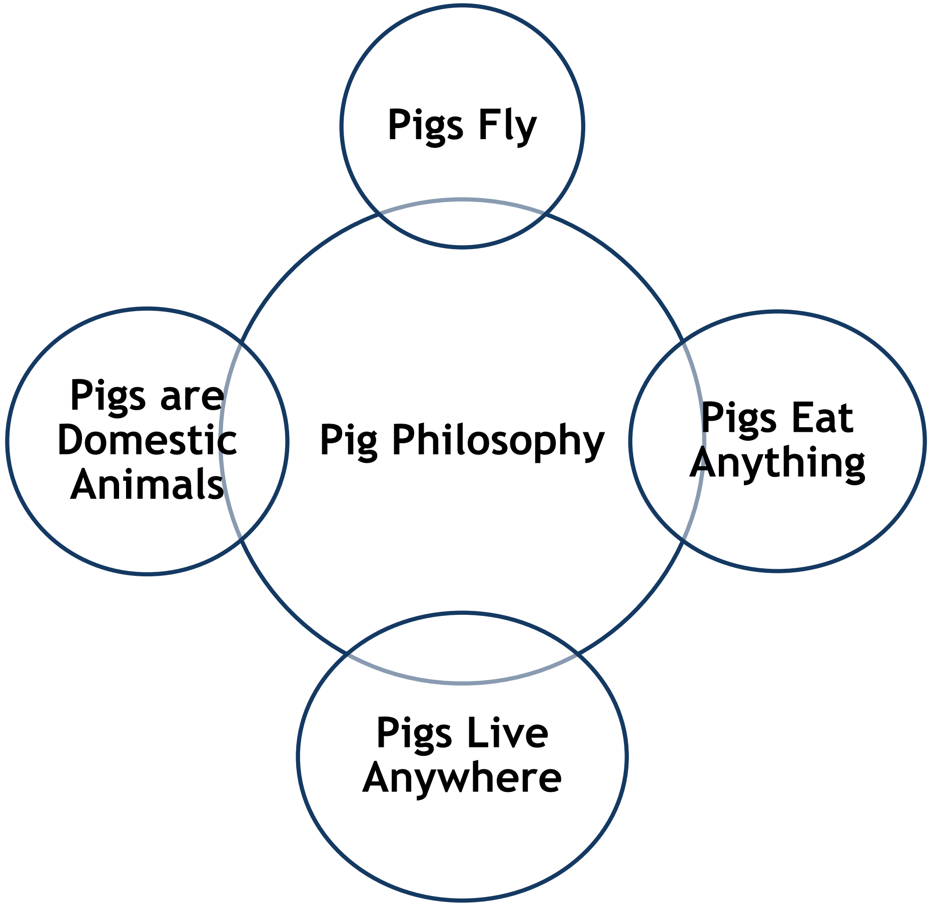
Pig on Hadoop

Pig on Hadoop

- Pig runs on Hadoop.
- Pig uses both Hadoop Distributed File System and MapReduce Programming.
- By default, Pig reads input files from HDFS. Pig stores the intermediate data (data produced by MapReduce jobs) and the output in HDFS.
- However, Pig can also read input from and place output to other sources.

Pig Philosophy

Pig Philosophy



Pig Latin Overview

Pig Latin Statements

Pig Latin Statements are generally ordered as follows:

1. **LOAD** statement that reads data from the file system.
2. Series of statements to perform transformations.
3. **DUMP** or **STORE** to display/store result.

```
A = load 'student' (rollno, name, gpa);

A = filter A by gpa > 4.0;


A = foreach A generate UPPER (name);
STORE A INTO ‘myreport’
```

Pig Latin Identifiers

Valid Identifier	Y	A1	A1_2014	Sample
Invalid Identifier	5	Sales$	Sales%	_Sales

Pig Latin Comments

In Pig Latin two types of comments are supported:

1. Single line comments that begin with “—”.
2. Multiline comments that begin with “/* and end with */”.

Operators in Pig Latin

Arithmetic	Comparison	Null	Boolean
+	= =	IS NULL	AND
-	! =	IS NOT NULL	OR
*	<		NOT
/	>		
%	<=		
	>=		

Data Types in Pig Latin

Simple Data Types

Name	Description
int	Whole numbers
long	Large whole numbers
float	Decimals
double	Very precise decimals
chararray	Text strings
bytearray	Raw bytes
datetime	Datetime
boolean	true or false

Complex Data Types

Name	Description
Tuple	An ordered set of fields. Example: (2,3)
Bag	A collection of tuples. Example: {(2,3),(7,5)}
map	key, value pair (open # Apache)

Running Pig

Running Pig

Pig can run in two ways:

1. Interactive Mode.

2. Batch Mode.

Execution Modes of Pig

Execution Modes of Pig

You can execute pig in two modes:

1. Local Mode.
2. Map Reduce Mode.

Relational Operators

Filter

Find the tuples of those student where the GPA is greater than 4.0.

```
A = load '/pigdemo/student.tsv' as (rollno:int, name:chararray, gpa:float);
B = filter A by gpa > 4.0;
DUMP B;
```

FOREACH

Display the name of all students in uppercase.

```
A = load '/pigdemo/student.tsv' as (rollno:int, name:chararray, gpa:float);

B = foreach A generate UPPER (name);

DUMP B;
```

Group

Group tuples of students based on their GPA.

```
A = load '/pigdemo/student.tsv' as (rollno:int, name:chararray, gpa:float);

B = GROUP A BY gpa;

DUMP B;
```

Distinct

To remove duplicate tuples of students.

```
A = load '/pigdemo/student.tsv' as (rollno:int, name:chararray, gpa:float);

B = DISTINCT A;

DUMP B;
```

Join

To join two relations namely, “student” and “department” based on the values contained in the “rollno” column.

```
A = load '/pigdemo/student.tsv' as (rollno:int, name:chararray, gpa:float);

B = load '/pigdemo/department.tsv' as (rollno:int, deptno:int,deptname:chararray);

C = JOIN A BY rollno, B BY rollno;

DUMP C;

DUMP B;
```

Split

To partition a relation based on the GPAs acquired by the students.

- GPA = 4.0, place it into relation X.
- GPA is < 4.0, place it into relation Y.

```
A = load '/pigdemo/student.tsv' as (rollno:int, name:chararray, gpa:float);

SPLIT A INTO X IF gpa==4.0, Y IF gpa<=4.0;

DUMP X;
```

Eval Function

Avg

To calculate the average marks for each student.

```
A = load '/pigdemo/student.csv' USING PigStorage (‘,’) as
(studname:chararray,marks:int);

B = GROUP A BY studname;

C = FOREACH B GENERATE A.studname, AVG(A.marks);

DUMP C;
```

Max

To calculate the maximum marks for each student.

```
A = load '/pigdemo/student.csv' USING PigStorage (‘,’) as (studname:chararray, marks:int);

B = GROUP A BY studname;

C = FOREACH B GENERATE A.studname, MAX(A.marks);

DUMP C;
```

Map

Map

MAP represents a key/value pair.

To depict the complex data type “map”.

```
John       [city#Bangalore]
Jack[city#Pune]
James      [city#Chennai]
```

```
A = load '/root/pigdemos/studentcity.tsv' Using PigStorage as
(studname:chararray,m:map[chararray]);

 B = foreach A generate m#'city' as CityName:chararray;

DUMP B
```

Piggy Bank

Piggy Bank

To use Piggy Bank upper function

```
register '/root/pigdemos/piggybank-0.12.0.jar';

A = load '/pigdemo/student.tsv' as (rollno:int, name:chararray, gpa:float);

upper = foreach A generate
 org.apache.pig.piggybank.evaluation.string.UPPER(name);

DUMP upper;
```

When to use Pig?

When to use Pig?

Pig can be used in the following situations:

1. When data loads are time sensitive.
2. When processing various data sources.
3. When analytical insights are required through sampling.

When NOT to use Pig?

Pig should not be used in the following situations:

1. When data is completely unstructured such as video, text, and audio.

2. When there is a time constraint because Pig is slower than MapReduce jobs.

Pig Vs. Hive

Pig Vs. Hive

Features	Pig	Hive
Used By	Programmers and Researchers	Analyst
Used For	Programming	Reporting
Language	Procedural data flow language	SQL Like
Suitable For	Semi - Structured	Structured
Schema / Types	Explicit	Implicit
UDF Support	YES	YES
Join / Order / Sort	YES	YES
DFS Direct Access	YES (Implicit)	YES (Explicit)
Web Interface	YES	NO
Partitions	YES	No
Shell	YES	YES

Answer a few quick questions …

Fill in the blanks

1. Pig is a ___________ language.
2. In Pig, ___________ is used to specify data flow.
3. Pig provides an ___________ to execute data flow.
4. ___________, ___________ are execution modes of Pig.
5. Pig is used in ___________ process.

Summary please…

Ask a few participants of the learning program to summarize the lecture.

References …

Further Readings

- http://pig.apache.org/docs/r0.12.0/index.html
- http://www.edureka.co/blog/introduction-to-pig/

Thank you