cct | College Dublin
Computing • IT • Business

Big Data Storage and Processing

MSc in Data Analytics

CCT College Dublin

Apache Pig

Week 8

Lecturer: Dr. Muhammad Iqbal*

Email: miqbal@cct.ie

Agenda

cct College Dublin
Computing • IT • Business

- Introduction to Apache Pig
- Apache Pig: Terminology and Operations
- Apache Pig: MapReduce and Execution Modes
- Apache Pig: Logical and Physical Plan
- Apache Pig: Simple & Complex Data Types and Input And Output
- Apache Pig: Relational Operators
- Apache Pig: User Defined Functions

Introduction to Apache Pig

cct College Dublin
Computing • IT • Business

- Yahoo created Apache Pig to simplify the process of mining Big datasets.
- Apache Pig provides a higher level of abstraction for processing Big datasets.
- It can be challenging to figure out how to fit your data processing into this pattern, which involves numerous **MapReduce stages**, despite the fact that MapReduce allows you to provide a **map** function followed by a **reduce** function.
- Apache Pig has much richer data structures, most of which are multivalued and nested and more powerful transformations. They include joins, for example, which are not considered as attractive in MapReduce. Pig is made up of two pieces
  - The language used to express data flows, called Pig Latin.
  - The execution environment to run Pig Latin programs. There are currently two environments: **local execution in a single JVM** and **distributed execution on a Hadoop cluster**.

Introduction to Apache Pig

- Pig Latin programs consist of multiple operations, or transformations, applied to input data to produce output.
- The operations describe a data flow that is translated into an executable representation and then executed by the Pig execution environment.
- Pig transforms the data into **a series of MapReduce jobs**, but as a programmer you are mostly unaware of this, so you are able to concentrate on the data instead of the execution.
- Pig is a scripting language for exploring Big datasets. One criticism of MapReduce is that the development cycle is very long.

Introduction to Apache Pig

- A **mapper** and **reducer** need to be **written**, **compiled** and **packaged**, **submitted**, and the results **retrieved** can be a lengthy process, and even with **Streaming**, which removes **compiling** and **packaging**, the experience would still be required.

- **A significant characteristic of Pig is its ability to process terabytes of data in response to a half-dozen lines of Pig Latin.**

- It is easy for a programmer to write a query in Pig, since it provides a number of commands that allow you to infer the data structures in your program.

- It enables easy data summarization, ad-hoc reporting and querying, and analysis of Big volumes of data.

- Pig interpreter runs on a client machine and there are not any administrative overhead required.

Apache Pig
Terminology

cct College Dublin
Computing • IT • Business

- Four major kinds of data is stored in Pig as mentioned below
  - An **Atom** is a simple data value - stored as a **string** but can be used as either a **string** or a **number**.
  - A **Tuple** is a data record consisting of a sequence of "**fields**".
    - Each **field** is a piece of data of any type (**atom, tuple or bag**)
  - A **Bag** is a set of **tuples** (also referred to as a 'Relation').
    - The concept of a "kind of a" **table.**
  - A **Map** is a map from keys that are string literals to values that can be any data type.
    - The concept of a **hash map.**

Apache Pig
Operations

cct College Dublin
Computing • IT • Business

- **Apache Pig provides the following operations**
  - Grouping and Joining
  - Combining and Splitting
  - Filtering
  - Aggregation
  - Sorting
- **Extensibility**
  - Support for **User Defined Functions (UDF's)**
- **Takes advantage of the same massive parallelism as native MapReduce**

UDF's

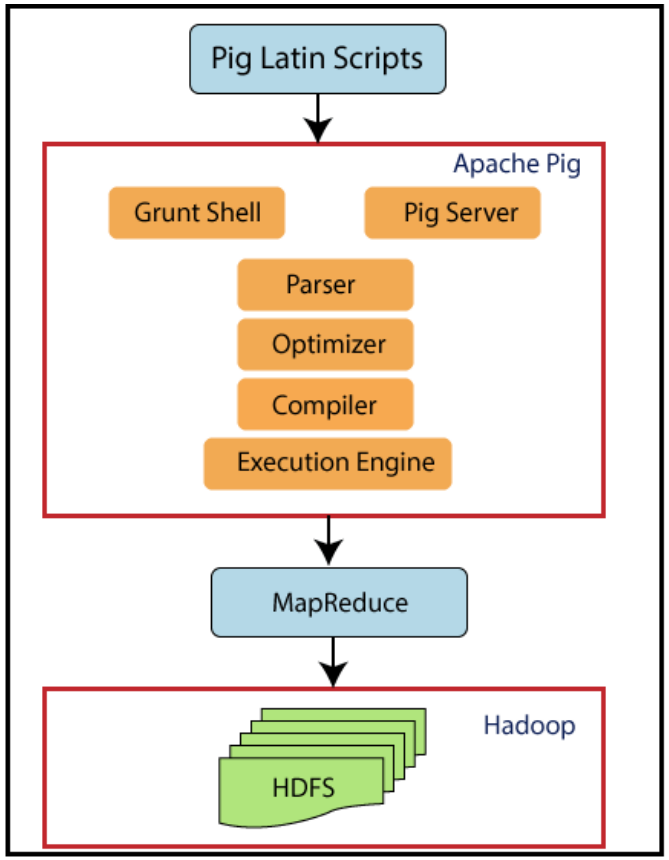
- UDF's are user defined functions and are of the following types:
  - EvalFunc
    - Used in the FOREACH clause
  - FilterFunc
    - Used in the FILTER by clause
  - LoadFunc
    - Used in the LOAD clause
  - StoreFunc
    - Used in the STORE clause

Apache Pig

MapReduce

cct College Dublin
Computing • IT • Business

- **Pig** is a client application
  - Requirement of a cluster is not essential
- Interprets **Pig** Latin scripts to **MapReduce** jobs
  - Parses Pig Latin scripts
  - Performs optimization
  - Performs compilation
  - Creates execution plan
- Submits **MapReduce** jobs to the **hdfs** cluster



Apache Pig

Execution Modes

cct | College Dublin
Computing • IT • Business

- Pig has two execution modes
  - **Local Mode** - all files are installed and run using your local host and file system
  - **MapReduce Mode** - all files are installed and run on a Hadoop cluster and HDFS installation
- **Interactive**
  - By using the **Grunt shell** by invoking Pig on the command line

```
$ pig -x local

grunt>
```

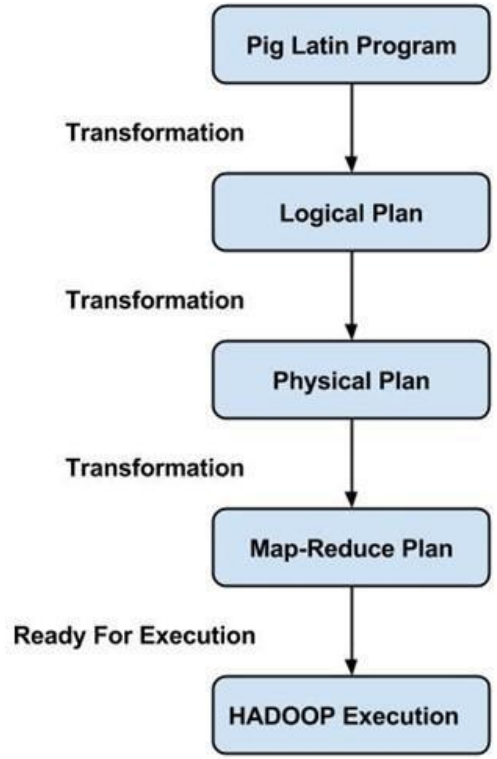
- **Batch**
  - Run Pig in batch mode using Pig Scripts and the "pig" command

```
$ pig -f id.pig -p <param>=<value> ...
```

Apache Pig
Logical and Physical Plan

cct College Dublin
Computing • IT • Business

- **Pig Latin scripts are generally organized as follows**
  - A **LOAD** statement reads data
  - A series of "**transformation**" statements process the data
  - A **STORE** statement writes the output to the filesystem
    - A **DUMP** statement displays output on the screen
- **Logical vs. physical plans**
  - All statements are stored and validated as a logical plan
  - Once a **STORE** or **DUMP** statement is found, the logical plan that will executed



Apache Pig

Sample Script

College Dublin
Computing • IT • Business

Input file

```
Mary had a little lamb
its fleece was white as snow
and everywhere that Mary went
the lamb was sure to go.
```

Pig Word Count Code

```
-- Load input from the file named Mary, and call the single
-- field in the record 'line'.
input = load 'mary' as (line);

-- TOKENIZE splits the line into a field for each word.
-- flatten will take the collection of records returned by
-- TOKENIZE and produce a separate record for each one, calling the single
-- field in the record word.
words = foreach input generate flatten(TOKENIZE(line)) as word;

-- Now group them together by each word.
grpd = group words by word;

-- Count them.
cntd = foreach grpd generate group, COUNT(words);

-- Print out the results.
dump cntd;
```

Output

```
Mary,2
had,1
a,1
little,1
lamb,2
its,1
fleece,1
was,2
white,1
as,1
snow,1
and,1
everywhere,1
that,1
went,1
the,1
sure,1
to,1
go,1
```

Apache Pig

"grunt" Shell

College Dublin
Computing • IT • Business

- Help is available

  `$ pig -h`

- Pig supports HDFS commands

  `grunt> pwd`

  - put, get, cp, ls, mkdir, rm, mv, etc.

- Pig Latin statements grouped together in a file
- Can be run from the command line or the shell
- Support parameter passing
- Comments are supported
  - Inline comments '--'
  - Block comments /* */

```
grunt> fs -ls /
Found 6 items
drwxr-xr-x   - cloudduggu supergroup          0 2020-07-01 03:08 /data
drwxr-xr-x   - cloudduggu supergroup          0 2020-06-29 04:57 /hive
drwxr-xr-x   - cloudduggu supergroup          0 2020-06-30 11:38 /home
drwxr-xr-x   - cloudduggu supergroup          0 2020-07-07 12:52 /pigdata
drwx-wx-wx   - cloudduggu supergroup          0 2020-06-29 21:27 /tmp
drwxr-xr-x   - cloudduggu supergroup          0 2020-06-30 07:39 /user
grunt>
```

Apache Pig

Simple & Complex Data Types

cct College Dublin
Computing • IT • Business

Type	Description
int	4-byte integer
long	8-byte integer
float	4-byte (single precision) floating point
double	8-byte (double precision) floating point
bytearray	Array of bytes; blob
chararray	String (“hello world”)
boolean	True/ False (case insensitive)
datetime	A date and time
biginteger	Java BigInteger
bigdecimal	Java BigDecimal

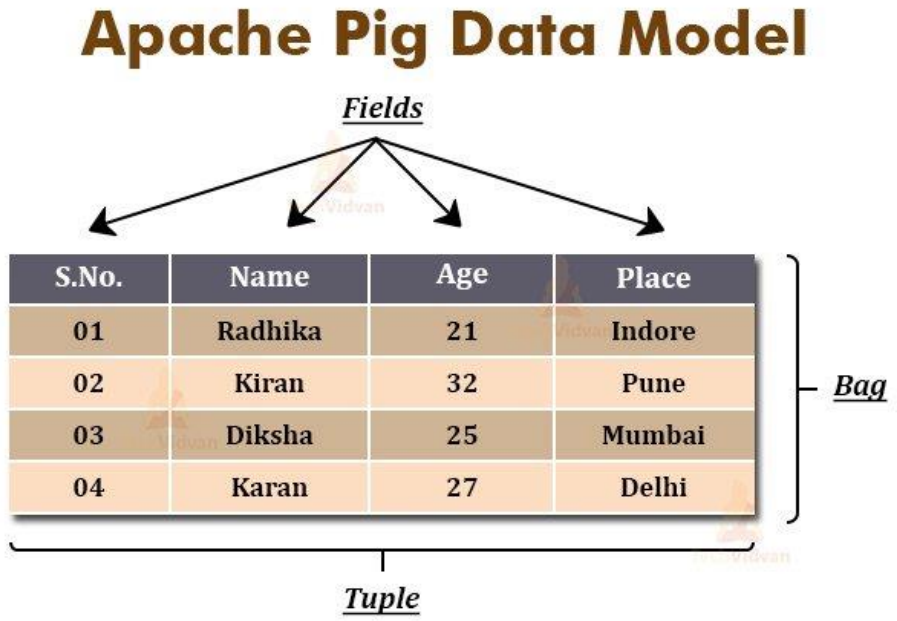
Complex Data Types

Type	Description
Tuple	Ordered set of fields (a “row / record”)
Bag	Collection of tuples (a “resultset / table”)
Map	A set of key-value pairs Keys must be of type chararray

Apache Pig
Data Formats

cct College Dublin
Computing • IT • Business

- **BinStorage**
  - Loads and stores data in machine-readable (binary) format
- **PigStorage**
  - Loads and stores data as structured, field delimited text files
- **TextLoader**
  - Loads unstructured data in **UTF-8 format**
- **PigDump**
  - Stores data in **UTF-8 format**
- **YourOwnFormat!**
  - via **UDFs**



Apache Pig
Data Loading Example

cct | College Dublin
Computing • IT • Business

- Loads data from an **HDFS** file

```
var = LOAD 'employees.txt';

var = LOAD 'employees.txt' AS (id, name, salary);

var = LOAD 'employees.txt' using PigStorage()
          AS (id, name, salary);
```

- Each **LOAD** statement defines a new bag
  - Each bag can have multiple elements (atoms)
  - Each element can be referenced by the name or position ($*n*)
- A bag is **immutable**
- A bag can be aliased and referenced later

Apache Pig
Input And Output

College Dublin
Computing • IT • Business

- **STORE**
  - Writes output to an **HDFS** file in a specified directory

```
grunt> STORE processed INTO 'processed_txt';
```

    - Fails if directory exists
    - Writes output files, part-[m|r]-xxxxx, to the directory
  - PigStorage can be used to specify a field delimiter
- **DUMP**
  - Write output to screen

```
grunt> DUMP processed;
```

Apache Pig
Relational Operators

cct College Dublin
Computing • IT • Business

- **FOREACH**
  - Applies expressions to every record in a bag
- **FILTER**
  - Filters by expression
- **GROUP**
  - Collect records with the same key
- **ORDER BY**
  - Sorting
- **DISTINCT**
  - Removes duplicates

Apache Pig
FOREACH . . .GENERATE

cct College Dublin
Computing • IT • Business

- Use the **FOREACH …GENERATE** operator to work with rows of data, call functions, etc.
- Basic syntax:

```
alias2 = FOREACH alias1 GENERATE expression;
```

- **Example:**

```
DUMP alias1;
```

```
(1,2,3) (4,2,1) (8,3,4) (4,3,3) (7,2,5) (8,4,3)
```

```
alias2 = FOREACH alias1 GENERATE col1, col2;
```

```
DUMP alias2;
```

```
(1,2) (4,2) (8,3) (4,3) (7,2) (8,4)
```

Apache Pig
FILTER. . .BY

College Dublin
Computing • IT • Business

- Use the **FILTER** operator to restrict tuples or rows of data
- Basic syntax:

```
alias2 = FILTER alias1 BY expression;
```

- **Example:**

```
DUMP alias1;
```

```
(1,2,3) (4,2,1) (8,3,4) (4,3,3) (7,2,5) (8,4,3)
```

```
alias2 = FILTER alias1 BY (col1 == 8) OR (NOT (col2+col3 > col1));
```

```
DUMP alias2;
```

```
(4,2,1) (8,3,4) (7,2,5) (8,4,3)
```

Apache Pig
GROUP. . .ALL

cct College Dublin
Computing • IT • Business

- Use the **GROUP…ALL** operator to group data
  - Use **GROUP** when only one relation is involved
  - Use **COGROUP** with multiple relations are involved
- **Basic syntax:**

```
alias2 = GROUP alias1 ALL;
```

- **Example:**

```
DUMP alias1;
(John,18,4.0F) (Mary,19,3.8F) (Bill,20,3.9F) (Joe,18,3.8F)
alias2 = GROUP alias1 BY col2;
DUMP alias2;
(18,{(John,18,4.0F),(Joe,18,3.8F)})
(19,{(Mary,19,3.8F)})
(20,{(Bill,20,3.9F)})
```

Apache Pig
ORDER. . .BY

cct College Dublin
Computing • IT • Business

- Use the **ORDER…BY** operator to sort a relation based on one or more fields
- **Basic syntax:**

```
alias = ORDER alias BY field_alias [ASC|DESC];
```

- **Example:**

```
DUMP alias1;
```

```
(1,2,3) (4,2,1) (8,3,4) (4,3,3) (7,2,5) (8,4,3)
```

```
alias2 = ORDER alias1 BY col3 DESC;
```

```
DUMP alias2;
```

```
(7,2,5) (8,3,4) (1,2,3) (4,3,3) (8,4,3) (4,2,1)
```

Apache Pig
Relational Operators

College Dublin
Computing • IT • Business

- **FLATTEN**
  - Used to un-nest tuples as well as bags
- **INNER JOIN**
  - Used to perform an inner join of two or more relations based on common field values
- **OUTER JOIN**
  - Used to perform left, right or full outer joins
- **SPLIT**
  - Used to partition the contents of a relation into two or more relations
- **SAMPLE**
  - Used to select a random data sample with the stated sample size

Category	Operator	Description
Loading and storing	LOAD	Loads data from the filesystem or other storage into a relation
	STORE	Saves a relation to the filesystem or other storage
	DUMP	Prints a relation to the console
Filtering	FILTER	Removes unwanted rows from a relation
	DISTINCT	Removes duplicate rows from a relation
	FOREACH...GENERATE	Adds or removes fields from a relation
	MAPREDUCE	Runs a MapReduce job using a relation as input
	STREAM	Transforms a relation using an external program
	SAMPLE	Selects a random sample of a relation
Grouping and joining	JOIN	Joins two or more relations
	COGROUP	Groups the data in two or more relations
	GROUP	Groups the data in a single relation
	CROSS	Creates the cross-product of two or more relations
Sorting	ORDER	Sorts a relation by one or more fields
	LIMIT	Limits the size of a relation to a maximum number of tuples
Combining and splitting	UNION	Combines two or more relations into one

https://www.w3schools.com/sql/sql_join_inner.asp

Apache Pig
User-Defined Functions

- **UDF** are written in Java, packaged as a jar file
  - Other languages include Jython, JavaScript, Ruby, Groovy, and Python
- Register the **jar** with the **REGISTER** statement
- Optionally, alias it with the **DEFINE** statement

```
REGISTER /src/myfunc.jar;

A = LOAD 'students';

B = FOREACH A GENERATE

myfunc.MyEvalFunc($0);
```

Macro is a kind of function written in Pig Latin.

- Pig Latin Macro and UDF Statements
- Enable to incorporate macros and user-defined functions into Pig scripts

Statement	Description
REGISTER	Registers a JAR file with the Pig runtime
DEFINE	Creates an alias for a macro, a UDF, streaming script, or a command specification
IMPORT	Import macros defined in a separate file into a script

Resources/ References

- https://pig.apache.org
- White,T., 2012. Hadoop:The definitive guide. "O'Reilly Media, Inc.". Thottuvaikkatumana, R., 2016.
- Some slides are used from the Boston University resources for learning purpose as mentioned as https://www.cs.bu.edu/faculty/gkollios/ada17/LectNotes/PigHive.pptx
- https://www.javatpoint.com/pig-udf
- https://www.cloudduggu.com/pig/
- Some images are used from Google search repository (https://www.google.ie/search) to enhance the level of learning.