International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878, Volume-8 Issue-5, January 2020

Implementation of ETL Process using Pig and Hadoop

Anushree Raj, Rio D. Souza

Abstract: ***ETL stands for extraction, transformation and loading, where extraction is done to active data from the source, transformation involve data cleansing, data filtering, data validation and finally application of certain rules and loading stores back the data to the destination repository where it has to finally reside. Pig is one of the most important to which could be applied in Extract, Transform and Load (ETL) process. It helps in applying the ETL approach to the large set of data. Initially Pig loads the data, and further is able to perform predictions, repetitions, expected conversions and further transformations. UDFs can be used to perform more complex algorithms during the transformation phase. The huge data processed by Pig, could be stored back in HDFS. In this paper we demonstrate the ETL process using Pig in Hadoop. Here we demonstrate how the files in HDFS are extracted, transformed and loaded back to HDFS using Pig. We extend the functionality of Pig Latin with Python UDFs to perform transformations.***

Keywords: ETL process, Extract, Load, HDFS ETL, Pig Latin, Python UDFs, Transform.

I. INTRODUCTION

Data warehouses are the repositories of data sets and information which are collected from various data sources with different formats and drives. Sophisticated tools are highly recommended to process large data sets since the data is expanding tremendously in terms of volume as well as variety. A specialized technique known as the ETL is a key process which enhances to collect all variety of data together in a standard, identical format [1]. We can implement the ETL process on the HDFS file using Pig. The data which is saved in the hadoop file system can be extracted into Pig for further data transformation. Pig Latin is one of the best scripting language to support the ETL process**.** It shows how to extract huge amount of data from a data source, transform the data so as to perform querying and also do the analysis jobs, and store back the end resultant data set onto a target destination database. Pig delivers high support to customize the processing functionalities through User Defined Functions (UDFs). Pig currently supports UDFs in six languages: Java, Jython, Python, JavaScript, Ruby, and Groovy. In this paper we demonstrate the ETL process using Pig in Hadoop. We perform extraction, transformation and loading of files from and store it back into the HDFS using Pig. The functionalities in Pig Latin is coded using Python UDFs to perform transformations.

Manuscript published on January 30, 2020.
* Correspondence Author
Anushree Raj*, Research Scholar, CSE Department, St Joseph Engineering College, Mangalore India. Email: anushree@stagnescollege.edu.in
Rio D'Souza Department of CSE, St Joseph Engineering College, Mangalore, India. Email: riod@sjec.ac.in

© The Authors. Published by Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP). This is an open access article under the CC-BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/)

II. ETL PROCESS

ETL is generally defined as a process which extracts the data from different RDBMS source systems, which further transforms the data that internally perform validations, manipulations and rule based corrections and finally loads back the data into the destination system. ETL which means Extract, Transform and Load provides a mechanism for the movement of data from various data source repositories into the destination databases.

Initially the data is mined from the source repository onto the specialized staging area where exactly the ETL process is performed. In the transformation stage, the data extracted from source repository is validated, corrected and processed. After the transformation stage lastly the data is loaded into the target destination database [2].

ETL Tools are normally used to rapidly import huge data sets to source database. While the actual process of computing, the extract, transform and load (ETL) denotes the process that involves:

- Extracting the datasets from an external sources
- Transforming these data to their functioning requirements
- Loading it back to the destination database

The initial stage of an ETL process comprises extracting the data from the any source repository. While at extraction phase, ETL attempts to retrieve the data from the source database, mine the significant data, and convert the data into a specified rule based format. It manipulates the data into a single format which is suitable for further transformation processing [3].

During the transformation phase a series of we defined rules or functions are applied to the extracted data from the source database to descend the data for loading into the destination repository. This phase is defined as the stage to clean and conform the provided information. The process of cleaning the data basically intends to correct the erroneous, invalid or irrelevant data and produces clean data for the decision makers or analyzers. Cleaning data also handles the missing data, rejects the irrelevant. The conforming of data intends to correction of data, which makes it compatible with existing data [4].

The load phase loads or stores back the data to wherever the data is to be sent into the destination database. ETL helps to improve efficiency of processing data since it codifies and reuses the data without any assistance of technical skills.

III. APACHE PIG

Pig is a high level scripting language that is used with Apache Hadoop. Pig plays a vital role in the ETL transaction model which describes on how

a process will extract data from a source, transform the data based on functional rule set and then load it into a destination database. Pig scripts are translated into a series of MapReduce jobs that are run on the Apache Hadoop cluster [5]. During the translation phase, the Pig interpreter performs optimizations in order to speed up the execution on Apache Hadoop. Pig is composed of two major parts: a high-level data flow language called Pig Latin, and an engine that parses, optimizes, and executes the Pig Latin scripts as a series of MapReduce jobs that are run on a Hadoop cluster. Pig is used to perform ETL jobs on Hadoop. It saves you from writing MapReduce code in Java while its syntax may look familiar to SQL users [6]. Pig is one of the easiest scripting language to write, understand, and maintain. It is a simple data transformation language which allows the processing of data to be defined as an ordered form of transformations. Pig is highly extensible which serves the best means to use the User Defined Functions (UDFs) which allow customization of processing the data, to be written in different languages, the one which we use in our implementation is Python.

IV. PYTHON UDFs IN PIG

A Pig UDF (User Defined Function) is a function that is accessible to Pig, but written in a language other than PigLatin. Pig allows you to register UDFs for use within a PigLatin script. There are three types of UDF function namely Eval UDFs, Aggregation UDFs and Filter UDFs [7]. Pig doesn't really perform the string manipulation or calculation functions, so we introduce an UDF that performs the deep functionalities. An aggregate UDF is a user defined function which is used to associate multiple pieces of information. Within a cluster of rows if at all only some portion of rows are to be used for analysis then it's advised to perform filter function. Python UDFs means C Python UDFs. It internally uses python command line to run the Python UDFs. The streaming of data is executed in and out of the python process. Pig allows to plug in to the Hadoop Streaming API. To run Python UDFs, Pig invokes the Python command line and streams data in and out of it.
The steps involved in invoking the python UDF is as follows:

i. Write a python script with a function to be called in Pig script.
ii. Add @outputSchema decorator to use it as a UDF.
iii. Within the Pig code Register the UDF using the REGISTER statement.
iv. Cal the registered UDF function using the alias using FOREACH and GENERATE statement.

V. IMPLEMENTATION

Apache Pig works on top of Hadoop. It is an analytical tool that analyzes large datasets that exist in the Hadoop File System [8]. In this paper we do the implementation using VMWare Workstation a virtual machine software which enhances to run multiple operating systems on a single physical host computer. We install the Hortanworks SandBox which is a single node HDP cluster running on virtual machine. It is used to implement the hadoop components which runs on only one node. We make use of WINSCP to transfer the HDFS file for execution. Two csv files Employee and Employee Detail are used to demonstrate the ETL process in Pig. In MapReduce mode, Pig reads data from HDFS and stores the results back in HDFS. We need to first generate the data in HDFS. To analyze data using Apache Pig, we have to initially load the data into Apache Pig from the HDFS [9].
The application of Pig in ETL process is one of the very efficient way of processing data in the HDFS. Since Pig is a SQL like scripting language it is not possible to have functionalities like data processing, string manipulation, calculations and many others. We can use UDFs defined in Python to assist most of the tasks in Pig. We save the data files from local file system to HDFS using copyFromLocal command. The extraction of data using Pig is done using a special LOAD statement. The LOAD operator is used to load data from the system into Pig. Pig contains many operators that enable complex transforming of data. The most common operators are FILTER, FOREACH, and GROUPBY. Python UDFs are an easy way of extending Pig's functionality and is also an easy way to transform and process large data. The FILTER operator works on tuples or rows of data. It selects tuples from a relation based on a condition. FOREACH operator works on columns of data and is similar to the SELECT statement in SQL. The GROUP operator groups together tuples that have the same group key into one or more relations. The STORE operator is used to execute previous Pig statements and store the results on the file system [10].

	A	B	C	D	E	F
1	EmpID	Firstnam	Gender	Email	Mobile	Home
2	101	Rahul	Male	rahul@ibm.com	9985647123	Jaipur
3	102	Aruna	Female	aruna@wipro.com	9758412324	Chochi
4	103	Manu	Male	manu@cognizent.co	8745896214	Manali
5	104	Senorita	Female	senorita@infosys.co	9845674112	Goa
6	105	Mohan	Male	mohan@amazon.co	7784522236	Bangalore
7	106	Rohan	Male	rohan@infosys.com	8885471256	Delhi
8	107	Suma	Female	suma@infosys.com	9874862145	Kanpur
9	108	Dinesh	Male	dinesh@wipro.com	8784562584	Udupi
10	109	Hamsa	Female	hamsa@wipro.com	7845222613	Magalore
11	110	Vani	Female	vani@amazon	8875489612	Chennai
12						

Fig 1: Employee data.csv

The sample data of an employee is stored in an Employee data file. The contents of Employee.csv are as shown in the Fig 1:

	A	B	C	D	E	F	G	H
1	EmpID	Firstname	Lastname	Gender	Age	Designatio	Location	Company
2	101	Rahul	Singh	Male	35	Team Lead	Bangalore	IBM
3	102	Aruna	Sai	Female	40	Manager	Chennai	Wipro
4	103	Manu	Reddy	Male	28	Software Er	Hydrabad	Cognizent
5	104	Senorita	Dspuza	Female	45	HR Manage	Goa	Infosys
6	105	Mohan	Benargi	Male	33	Analyst	Bombay	Amazon
7	106	Rohan	Sham	Male	42	Manager	Bangalore	Infosys
8	107	Suma	Rai	Female	32	Software Er	Goa	Infosys
9	108	Dinesh	Karkera	Male	38	Team Lead	Chennai	Wipro
10	109	Hamsa	Kamath	Female	29	Software Er	Hydrabad	Wipro
11	110	Vani	Jayram	Female	33	Data Engine	Bangalore	Amazon

Fig 2 Employee details data.

The sample data of an employee's additional information is stored in an Employee details data file. The contents of Employee_details are as shown in the Fig 2:

```
Job DAG:
job_1571981400320_0058


2019-10-25 12:47:13,358 [main] INFO  org.apache.pig.backend.hadoop.executionengi
ne.mapReduceLayer.MapReduceLauncher - Success!
2019-10-25 12:47:13,359 [main] INFO  org.apache.hadoop.conf.Configuration.deprec
ation - fs.default.name is deprecated. Instead, use fs.defaultFS
2019-10-25 12:47:13,371 [main] INFO  org.apache.hadoop.mapreduce.lib.input.FileI
nputFormat - Total input paths to process : 1
2019-10-25 12:47:13,371 [main] INFO  org.apache.pig.backend.hadoop.executionengi
ne.util.MapRedUtil - Total input paths to process : 1
(EmpID,Firstname,Lastname,Gender,Age,Designation,Location,Company)
(101,Rahul,Singh,Male,35,Team Lead,Bangalore,IBM)
(102,Aruna,Sai,Female,40,Manager,Chennai,Wipro)
(103,Manu,Reddy,Male,28,Software Engineer,Hydrabad,Cognizent)
(104,Senorita,Dspuza,Female,45,HR Manager,Goa,Infosys)
(105,Mohan,Benargi,Male,33,Analyst,Bombay,Amazon)
(106,Rohan,Sham,Male,42,Manager,Bangalore,Infosys)
(107,Suma,Rai,Female,32,Software Engineer,Goa,Infosys)
(108,Dinesh,Karkera,Male,38,Team Lead,Chennai,Wipro)
(109,Hamsa,Kamath,Female,29,Software Engineer,Hydrabad,Wipro)
(110,Vani,Jayram,Female,33,Data Engineer,Bangalore,Amazon)
grunt> _
```

Fig 3: Display Employee.csv in HDFS

The data is extracted from the HDFS using copyFromLocal command and displayed using Pig LOAD statement as shown in the Fig 3.

```
Job DAG:
job_1571981400320_0063


2019-10-25 13:04:46,981 [main] INFO  org.apache.pig.backend.hadoop.executionengi
ne.mapReduceLayer.MapReduceLauncher - Success!
2019-10-25 13:04:46,983 [main] INFO  org.apache.hadoop.conf.Configuration.deprec
ation - fs.default.name is deprecated. Instead, use fs.defaultFS
2019-10-25 13:04:46,995 [main] INFO  org.apache.hadoop.mapreduce.lib.input.FileI
nputFormat - Total input paths to process : 1
2019-10-25 13:04:46,995 [main] INFO  org.apache.pig.backend.hadoop.executionengi
ne.util.MapRedUtil - Total input paths to process : 1
(EmpID,Firstname,Log_Hours,Log_Days,Mobile,Home)
(101,Rahul,300,15,9985647123,Jaipur)
(102,Aruna,400,18,9758412324,Chochi)
(103,Manu,218,12,8745896214,Manali)
(104,Senorita,430,16,9845674112,Goa)
(105,Mohan,504,20,7784522236,Bangalore)
(106,Rohan,600,28,8885471256,Delhi)
(107,Suma,560,25,9874862145,Kanpur)
(108,Dinesh,480,24,8784562584,Udupi)
(109,Hamsa,350,18,7845222613,Magalore)
(110,Vani,280,14,8875489612,Chennai)
grunt> _
```

Fig 4: Display Emp_Detail.csv in HDFS

The file contents of employee's information file which contains Employee details is displayed using Pig Load command is showed in Fig 4.

```
2019-10-25 13:06:06,962 [main] WARN  org.apache.pig.backend.hadoop.executionengi
ne.mapReduceLayer.MapReduceLauncher - Encountered Warning FIELD_DISCARDED_TYPE_C
ONVERSION_FAILED 1 time(s).
2019-10-25 13:06:06,962 [main] INFO  org.apache.pig.backend.hadoop.executionengi
ne.mapReduceLayer.MapReduceLauncher - Success!
2019-10-25 13:06:06,963 [main] INFO  org.apache.hadoop.conf.Configuration.deprec
ation - fs.default.name is deprecated. Instead, use fs.defaultFS
2019-10-25 13:06:06,963 [main] WARN  org.apache.pig.data.SchemaTupleBackend - Sc
hemaTupleBackend has already been initialized
2019-10-25 13:06:06,966 [main] INFO  org.apache.hadoop.mapreduce.lib.input.FileI
nputFormat - Total input paths to process : 1
2019-10-25 13:06:06,966 [main] INFO  org.apache.pig.backend.hadoop.executionengi
ne.util.MapRedUtil - Total input paths to process : 1
(101,Rahul,300,15,9985647123,Jaipur)
(102,Aruna,400,18,9758412324,Chochi)
(103,Manu,218,12,8745896214,Manali)
(104,Senorita,430,16,9845674112,Goa)
(105,Mohan,504,20,7784522236,Bangalore)
(106,Rohan,600,28,8885471256,Delhi)
(107,Suma,560,25,9874862145,Kanpur)
(108,Dinesh,480,24,8784562584,Udupi)
(109,Hamsa,350,18,7845222613,Magalore)
(110,Vani,280,14,8875489612,Chennai)
grunt> _
```

Fig 5: Display details where EmpID greater than equal to 1 using Pig script

The contents of the Employee file using Filter command after using the Python UDF to delete the missing data rows in Pig is showed in Fig 5.

```
2019-10-25 12:56:32,452 [main] WARN  org.apache.pig.backend.hadoop.executionengi
ne.mapReduceLayer.MapReduceLauncher - Encountered Warning FIELD_DISCARDED_TYPE_C
ONVERSION_FAILED 1 time(s).
2019-10-25 12:56:32,452 [main] INFO  org.apache.pig.backend.hadoop.executionengi
ne.mapReduceLayer.MapReduceLauncher - Success!
2019-10-25 12:56:32,453 [main] INFO  org.apache.hadoop.conf.Configuration.deprec
ation - fs.default.name is deprecated. Instead, use fs.defaultFS
2019-10-25 12:56:32,453 [main] WARN  org.apache.pig.data.SchemaTupleBackend - Sc
hemaTupleBackend has already been initialized
2019-10-25 12:56:32,456 [main] INFO  org.apache.hadoop.mapreduce.lib.input.FileI
nputFormat - Total input paths to process : 1
2019-10-25 12:56:32,456 [main] INFO  org.apache.pig.backend.hadoop.executionengi
ne.util.MapRedUtil - Total input paths to process : 1
(101,Rahul,Male,rahul@ibm.com,9985647123,Jaipur)
(102,Aruna,Female,aruna@wipro.com,9758412324,Chochi)
(103,Manu,Male,manu@cognizent.com,8745896214,Manali)
(104,Senorita,Female,senorita@infosys.com,9845674112,Goa)
(105,Mohan,Male,mohan@amazon.com,7784522236,Bangalore)
(106,Rohan,Male,rohan@infosys.com,8885471256,Delhi)
(107,Suma,Female,suma@infosys.com,9874862145,Kanpur)
(108,Dinesh,Male,dinesh@wipro.com,8784562584,Udupi)
(109,Hamsa,Female,hamsa@wipro.com,7845222613,Magalore)
(110,Vani,Female,vani@amazon,8875489612,Chennai)
grunt> _
```

Fig 6: Display Emp_Detail Log using Pig script

The contents of the Employee_detail file using FILTER after using the Python UDF to delete the null value rows in Pig is showed in Fig 6.

```
2019-10-25 12:51:27,270 [main] WARN  org.apache.pig.backend.hadoop.executionengi
ne.mapReduceLayer.MapReduceLauncher - Encountered Warning FIELD_DISCARDED_TYPE_C
ONVERSION_FAILED 1 time(s).
2019-10-25 12:51:27,270 [main] INFO  org.apache.pig.backend.hadoop.executionengi
ne.mapReduceLayer.MapReduceLauncher - Success!
2019-10-25 12:51:27,270 [main] INFO  org.apache.hadoop.conf.Configuration.deprec
ation - fs.default.name is deprecated. Instead, use fs.defaultFS
2019-10-25 12:51:27,271 [main] INFO  org.apache.pig.data.SchemaTupleBackend - Ke
y [pig.schematuple] was not set... will not generate code.
2019-10-25 12:51:27,274 [main] INFO  org.apache.hadoop.mapreduce.lib.input.FileI
nputFormat - Total input paths to process : 1
2019-10-25 12:51:27,275 [main] INFO  org.apache.pig.backend.hadoop.executionengi
ne.util.MapRedUtil - Total input paths to process : 1
(101,Rahul,Team Lead)
(102,Aruna,Manager)
(103,Manu,Software Engineer)
(104,Senorita,HR Manager)
(105,Mohan,Analyst)
(106,Rohan,Manager)
(107,Suma,Software Engineer)
(108,Dinesh,Team Lead)
(109,Hamsa,Software Engineer)
(110,Vani,Data Engineer)
grunt> _
```

Fig 8: Display information of employee using Pig script

The selective details of employee information is displayed using Filter command and perform data corrections by executing python UDF in Pig, is showed in Fig 8.

```
2019-10-25 13:10:08,418 [main] WARN  org.apache.pig.backend.hadoop.executionengi
ne.mapReduceLayer.MapReduceLauncher - Encountered Warning FIELD_DISCARDED_TYPE_C
ONVERSION_FAILED 1 time(s).
2019-10-25 13:10:08,418 [main] INFO  org.apache.pig.backend.hadoop.executionengi
ne.mapReduceLayer.MapReduceLauncher - Success!
2019-10-25 13:10:08,418 [main] INFO  org.apache.hadoop.conf.Configuration.deprec
ation - fs.default.name is deprecated. Instead, use fs.defaultFS
2019-10-25 13:10:08,419 [main] INFO  org.apache.pig.data.SchemaTupleBackend - Ke
y [pig.schematuple] was not set... will not generate code.
2019-10-25 13:10:08,423 [main] INFO  org.apache.hadoop.mapreduce.lib.input.FileI
nputFormat - Total input paths to process : 1
2019-10-25 13:10:08,423 [main] INFO  org.apache.pig.backend.hadoop.executionengi
ne.util.MapRedUtil - Total input paths to process : 1
(101,300,15)
(102,400,18)
(103,218,12)
(104,430,16)
(105,504,20)
(106,600,28)
(107,560,25)
(108,480,24)
(109,350,18)
(110,280,14)
grunt> _
```

Fig 9: Display the working hours log details.

The working hours log details from Employee detail file by executing a function to output only the specified fields using python UDF in Pig is showed in Fig 9.

Fig 10: Display the output of GroupByEmpID

The logging hours and the days logged is displayed by grouping the two files and retrieving the contents using Groupby EmpID is showed in Fig 10.

Fig 10: Join data from Employee and Emp_Detail csv files.

The contents of Employee and the Employee details is joined with respect to the empID and displayed using Join command in Pig is shown in Fig 10.

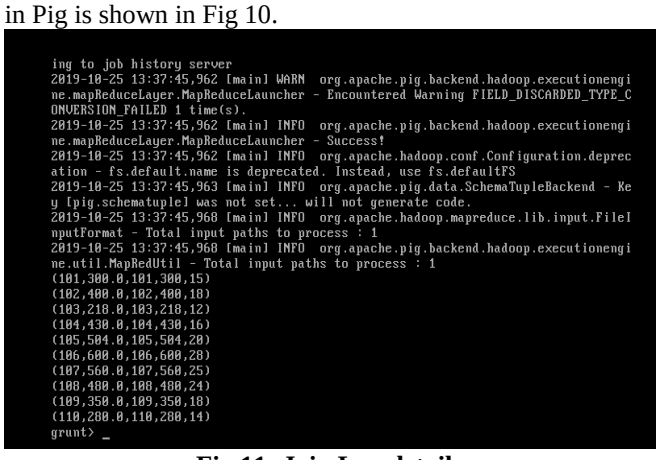

Fig 11: Join Log details

The details of empId, Logging hours and logged days of Employee and Employee is joined as shown in the Fig 11. Using the Join command in Pig.

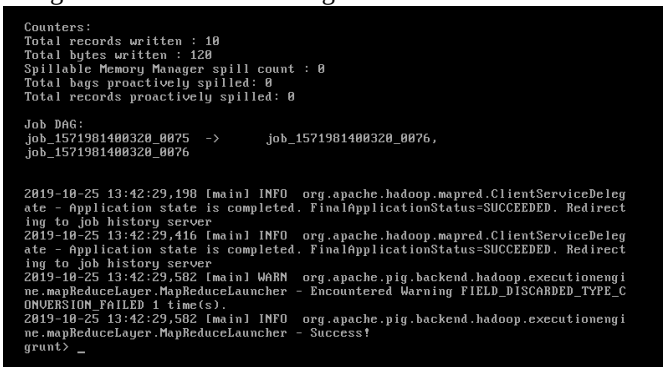

Fig 12: Store the output in HDFS

The desired output is successfully stored in HDFS after all the ETL process as shown in the Fig 12.

VI. RESULT AND DISCUSSION

The experiment was conducted for different csv data files of varying sizes. The large data sets in the file system enables huge processing time. We used Hadoop architecture to implement MapReduce based application to process on huge data files. The file was extracted from the Hadoop File system and performed the transformation operations using Pig. Python UDFs del_null(), str_validate(), mobile_validate(), check_missdata(), group_log(), output_str(), extratct_id() and filter_log() were used to support the data transformation and string manipulation functions in Pig Latin. The output file was then loaded back the HDFS after the processing.

V. CONCLUSION

Extraction, Transformation, and Loading (ETL) processes are the mail tools used for the processes taking place in any data warehouse architecture. The data entering into the warehouse needs to undergo the ETL process at the staging area. In this research paper, the entire process of implementation of ETL using Pig and Hadoop was demonstrated. The external source data file was copied to the HDFS, then loaded into the Pig and ETL process was performed using Pig Latin and Python UDFs. The output file after the transformations and manipulations was further stored back into the HDFS. The work can be further extended to perform ETL process on various advanced functional aspects of data processing techniques.

REFERENCES

1. P. Vassiliadis, A. Simitsis, and S. Skiadopoulos. Conceptual modeling for ETL processes. In Proceedings of the ACM 5th International Workshop on Data Warehousing and OLAP (DOLAP '02), pages 14–21, 2002.
2. http://en.wikipedia.org/wiki/Extract,_transform,_load
3. Anastasios Karagiannis, Panos Vassiliadis, Alkis Simitsis. Macro Level Scheduling of ETL Workflows. In 9th International Workshop on Quality in Databases (QDB 2011), in Conjunction with VLDB, 2011.
4. Thomas Jorg and Stefan Dessloch. Formalizing ETL Jobs for Incremental Loading of Data Warehouses. In Datenbanksysteme in Business, Technologie Und Web (BTW), 13.Münster, Germany, 2009
5. https://www.guru99.com/introduction-to-pig-and-hive.html
6. Hadoop with Python by Zachary Radtka and Donold Miner - O'Reilly Media
7. Python for Data Analysis by Wes McKinney
8. https://www.tutorialspoint.com/apache_pig
9. https://pig.apache.org/docs/r0.16.0/udf.html