An Insight Look Into Pig and Its Implementation

Surajit Mohanty[1], Sameer Kumar Das[2], Himanshu Suman[3], Piyush Maharana[4], Raman Ratnakar[5]
[1-2]Asst. professor, [3-5]B_Tech Scholar
[1, 3-5]Department of Computer Science Engineering, DRIEMS, Cuttack, India
[2]Department of Computer Science & Engineering, GATE, Berhampur, India

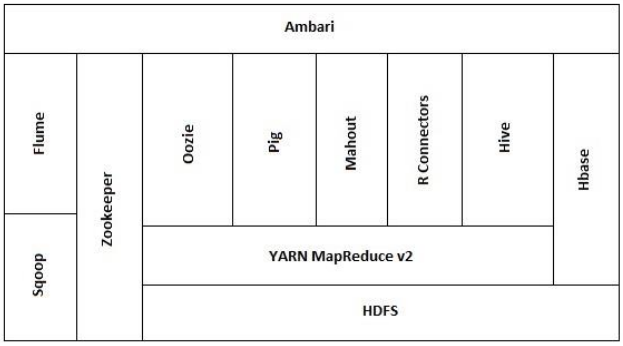
Abstract – Big Data is the term used now a days for referring to the data sets that are large and complex. The applications used for processing traditional data are inadequate. Now Apache Hadoop is a platform that is used for processing of big data. Hadoop has several tools that facilitate the processing of complex data like hive, mahout, zookeeper, pig and others. Each of these tools cover the different challenges faced by the users. We focused on Pig which is a scripting language for exploring large data sets. It has the ability to process terabytes of data for a few lines of code. All parts of processing path are customizable by the user, storing, filtering, grouping and joining. We discuss about Pig in brief its features and entities. Then we take up a sample problem using a dataset and generate the results. So that we can create a contrast between Pig and Traditional data processing applications. We explain about how we load and store complex information (dataset) and processing different kind of queries and running scripts in Pig.

Index Terms – Big Data, Pig, Filtering, Grouping, Joining.

1. INTRODUCTION

Apache Hadoop ecosystem consists of several parts i.e. consisting of frameworks and tools [fig.1].

- Distributed processing framework (YARN MapReduce V2)
- Hadoop Distributed File System(HDFS)
- Provisioning managing and monitoring of Hadoop clusters tool (Ambari).
- Workflow tool (Oozie).
- Scripting tool (Pig).
- Machine learning tool (Mahout).
- Statistics tool(R connector).
- SQL query tool (HIVE).
- Columnar store tool (Hbase).
- Coordination tool (Zookeeper).
- Log collector (Flame).
- Data exchange(Sqoop)



[Fig.1 Apache Hadoop ecosystem]

Apache Pig is a high level platform used with Hadoop for creation of MapReduce program. It raises the level of abstraction for processing large database. In MapReduce one can specify a map function followed by a reduce function. But with Pig the data structures become richer, typically being multivalued and nested. Such as 'joins' which are not possible in MapReduce.

2. RELATED WORK

Pig is a high level scripting language that is used with Apache Hadoop. Pig enables data workers to write complex data transformations without knowing Java. Pig's simple SQL-like scripting language is called Pig Latin, and appeals to developers already familiar with scripting languages and SQL. [2]

It has two components:

- Pig latin is the language used to express dataflow.
- Pig latin programs are run in two environments named
  - Local execution in single JVM
  - Distributed execution in Hadoop cluster.

It is a scripting language for exploring large data sets. It has the ability to process terabytes of data for a few lines of code. All parts of processing path are customizable by the user, storing, filtering, grouping and joining. Pig runs as a client side application. It has two nodes i.e. local node and MapReduce mode.

Local node

Local nodes are suitable only for small data sets. In this mode Pig runs in a single JVM and access the local file system.

Map Reduce mode

Here queries are translated into MapReduce job end one run on a Hadoop cluster. It is applicable for large datasets. Here Pig checks the HADOOP_HOME environment variable for which client to run.

3 WAYS FOR RUNNING PIG PROGRAMS

a) Script : Pig can a script file containing pig commands
b) Grunt: By default Pig is to run this program. It is an infective shell for running Pig commands.
c) Embedded: pig programs in java can be run using Pig server classes, JDBC for running sql.

Comparison with database

PIG	SQL
Data flow programming layer	Declarable programming layer
Here each step is a single transformation.	All the sets of constraints together define the output.
Schema can be defined at runtime.	Tightly predefined schema.
Supports complex and nested Data Structure.	Operates on single data structure.

Pig Latin

A Pig latin program consists of a collection of statements. These statements might be an operation or a command. Statement usually terminate with a semicolon.

E.g.-

grouped_records= GROUP records by year;

Here, operators and commands are not case sensitive, whereas aliases and function names are case sensitive. When a pig is executed each statement is passed wholly. The interpreter makes a logical plan for the program so far.

The DUMP statement is the execution trigger until this all statement are planned and added and at last they are all compiled into a physical plan and executed. [1]

Brief Overview

1. Relational Operator[3]:
   - Loading and storing –LOAD, STORE, DUMP.
   - Filtering – FILTER, DISTINCT, FOREACH, MAPREDUCE, STREAM.
   - Grouping and joining – JOIN, COGROUP, GROUP, CLOSE.
   - Sorting – ORDER, RANK, LIMIT.
   - Combining and splitting – UNION, SPLIT.
2. Diagnostic operator
   DESCRIBE, EXPLAIN, ILLUSTRATE
3. Macro and OPF
   REGISTER, DEFINE, IMPORT

Commands

- HDFS – cat, cd, cp, fs, ls, mkdir, mv.
- Map Reduce – kill
- Utility – clear, exec, help, history, quit(\q), run, set

Pig programs always run in JVM. However one can use other language such as Python and Java script also.

Expressions

Constant, Field, Projection, Map lockup, cost, Arithmetic, Conditional, Comparison, Boolean, Flatter.

Types

Boolean, Numeric, Text, Binary, Temporal, Complex

Functions

- Eval: A function that takes one or more expression and returns another expressions.
- Filter: A special type of eval function that returns a logical Boolean result.
- Load: A function that specifies how to load data into a relational from external storage.
- Store: A function those specifies how to save contents of a relation to external storage.

Macros

Like in other language, Macros are also provided for reusing pieces of code within Pig Latin itself. At runtime, Pig expands the macro using the macro definition exactly.

User Defined Function

Designers can write custom codes in pig like in any other scripting language. Normally these are written in Java. Now to use a new function, it has to be compiled and packaged into a JAR file.

Dynamic invokers allow calling method directly from pig script but repeated call for a large data set can result in significant overload.

Like Hadoop, in Pig also data loading begin before the mapper runs.so it is extremely important to split the input such that it can be handled independently by each mapper [4].

Filtering the income data

Now with the huge amount of data flowing in. Filtering the irrelevant data is crucial, so, as to improve efficiency.

FOREACH GENERATOR

This operator has a nested form to support more complex processes. It can be used to remove fields or to generate new ones.

STREAM

This operator allows transforming data in a relation using an external program or script. It uses Pig Storage to serialize and desterilized relations to and from the program's standard input and output stream.

Grouping and Joining Data

Joining data requires expensive work in Map Reduce but Pig provides built in functions.

JOIN AND COGROUP

JOIN gives a set of tuple, while COGROUP create a nested set of output tuples [5].

Some practices to be adhered to:-

- ✓ *Parallelism*
  While in MapReduce mode the degree of parallelism must match the size of the data set.
- ✓ *Anonymous relation*

Instead of defining a name for every relation. Pig allows applying the diagnostic operator like DUMP or DESCRIBE to latest defined relation.

- ✓ *Parameter substitution*

If a Pig script is called regularly (i.e. with different parameters) we can pass the parameter during the runtime, using –param and –param –file option.

- ✓ *Dynamic parameters*

Pig even allows the parameters passed in the form of commands and scripts. Backtick supper fix a great feature. With this support parameters can be defined in the similar way in file or command line.

- ✓ *Parameter substitution processing*

It is a preprocessing step before the script is run. Substitution happen in two modes:

- Data run mode
- Normal mode

- ✓ *Performance consideration in JOIN*

JOIN is one of the key advantages of Pig after Map Reduce. If one of the datasets is small enough to fit into memory, a replicated JOIN is using likely to provide better performance. [3]

4. PORPOSED MODELLING

Getting the data into the required shape requires information about data so that he data can be visualized. E.g. data generated from social media sources, the user will require general sense as customers searching for a particular set of products and the user will have to understand what it is and has to visualize it. Without some sort of context, visualization tools are likely to be of less value to the user. A solution to the challenge is having the proper domain expertise in place. It means analyzers of the data must have a deep understanding of where the data comes from, who will consume the data and how will they interpret the information. Even if we can find and analyze data quickly and put it in the proper context for the target people who will be using the information, value of data generated for decision-making purposes will be compromised if the data generated is inaccurate or untimely. This is the challenge faced by data analysts, but taking into account the voluminous amount of information involved in big data projects Again, data visualization will only prove to be a valuable tool if the data quality is assured. Analysis becomes difficult while extremely large amounts of information or a variety of categories of information is involved. One way of resolving the problem is to cluster data into a higher-level view so that smaller groups of data become visible. By grouping the data together, or "binning," data can be more effectively visualized.

In the following problem we used a publicly available million song dataset which is a freely-available collection of audio features and metadata for a million contemporary popular music tracks. We worked upon it using Pig tool to get solution to certain problems such as loading and storing the data, finding the song density, filtering the data.

Solution Approach

As we are focusing on Pig tool which basically works upon scripting.

Problem-1

Loading and storing the million song dataset.

Solution:

To work upon any dataset we must load the data and define the location for storing the output generated by the user scripts/queries.

Syntax for loading the data

1. var1 = LOAD 'input location' USING PigStorage('\t') AS (data_name1:datatype, data_name2 :datatype,……);

Syntax for storing the data

2. STORE var1 INTO 'output location' USING PigStorage('\t');

Once we have defined the loading and storing location of the dataset and the output respectively, we can proceed with our queries.

E.g. suppose we have to filter the data.

SYNTAX:

1. var2 = FILTER var1 BY data_name value;

Example

2. filtered_songs = FILTER songs BY artisthotness > 0.5;

Our aim was to find the top 50 songs with highest sound densities. So we will load and define the storing location first. Then

1. Filtered_songs = Filter songs by duration>0;
2. Song_density = FOREACH filtered_songs GENERATE artist_name, title, density(segments_start, duration);
3. density_ordered = ORDER song_density by density DESC;
4. top_density = LIMIT density_ordered 50;

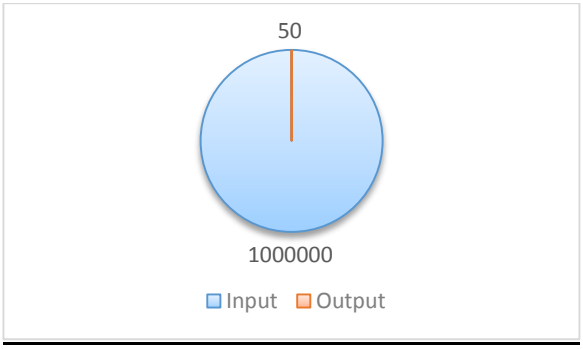
Output generated:

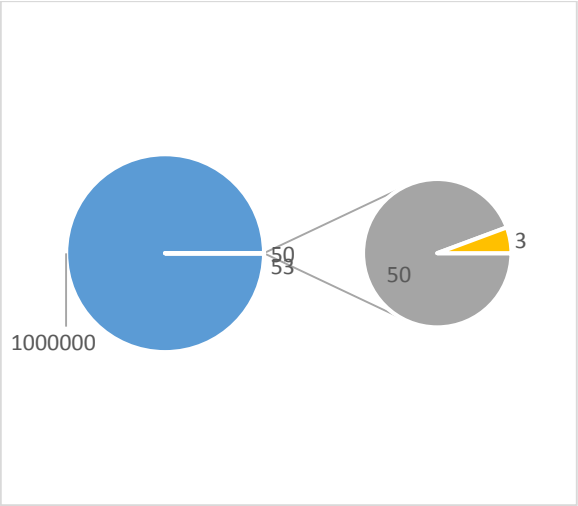
```
Olivia Newton-John      Rainy Days And Mondays  12.778736183
The George Benson Quartet       Clockwise       10.9511033237
Pepe Garsia     Get Down        9.99400359784
Blir    2:12    9.63832706439
Helmut Zacharias - Geige Violin - And His Orchestra     Dark Eyes  -  Schwarze Augen    9.58060884769
Las Pajarillas  El Muchacho Alegre (Album Version)      9.58060884769
Ryuji Takeuchi  Ichigo Ichie    9.27441307199
Strojovna 07    129     9.24124916507
Makaton The Feeding Circle      9.14132023696
Grave Plott     Street Life featuring Spice-1   9.13063466151
Marco V Savage  9.11943769254
Basic Channel   Enforcement      9.05111641777
Makaton I am the Game   9.04812041233
Strojovna 07    Ezachrum        9.04577978154
Takaaki Itoh    Onset   9.04379313976
Dare and Haste  Tribal Masochism        9.02938620304
Zeta Funk       Se Sensation    9.01652277799
Fatals Picards  Bernard Lavilliers [En Public]  9.01652277799
MarilÃ³ y Lourdes GarcÃa        Palmas dobladas 300 pm  9.00271385372
The Advent      GROUND FORCE    8.97223880602
Space DJ's      AK-47   8.93757562984
DJ ReMix Factory        The Promise (As Made Famous by When in Rome) (ReMixed)  8.93746329256
DJ ReMix Factory        The Promise (As Made Famous by When in Rome) (NRG Remix)        8.93746329256
Exium   Ethereal       8.91976788586
Ryoji Ikeda     +.      8.89347211662
Andreas Kauffelt        Heat    8.87202465263
Axel Karakasis  Evolved 8.86633618956
Strojovna 07    Fiesta  8.86490365308
Lars Klein      Gaikokujin      8.84033270335
Kike Pravda     Kike Pravda - Abl       8.83479607684
Chris Liebing   Art     8.82503370246
Strojovna 07    Ja a Vlasto     8.82432558209
Boris Divider   Remote Operator 8.81087806769
Fanon Flowers   Mr Sys  8.80350820586
James Ruskin    Logical Force   8.79814142252
Dare and Haste  Tribal Masochism Retake 8.78024694149
Concrete Djz    Encrypter B22   8.76423040134
Concrete Djz    Encrypter B22   8.76423040134
Mark Verbos     One Track Mind  8.76296474969
Raul Mezcolanza Guay    8.7624072248
Axel Karakasis  Raindrops       8.762130005
James Ruskin    Logical Force   8.7487498822
Inigo Kennedy   Devastator      8.72735289132
Cristian Varela Lesbian I Tunes 8.72228026405
Fanon Flowers   Untitled        8.7105820664
Jeff Mills      The Hacker      8.70891293817
Joey Beltram    Ball Park       8.70886677716
James Ruskin    Indirect World  8.70648475581
James Ruskin    Solex   8.70528942906
Axel Karakasis  Average 8.69804009675
```

5. RESULT ANALYSIS

The input provided amount to a million. Variable Song_density has a million fields each having 3 data. Then they were ordered in descending order. And the no of output was limited to 50.

Input –Output Pie chart





Data fields produced comparison

6. CONCLUSION

As Big data invades into more and more fields the data becomes complex. It becomes incredibly different to process data, with on hand management tools/traditional data processing application. The challenges are memory, namely, capturing, storing, searching, sorting, transferring, analyzing and visualizing. The trend of big data is picking up in the market. Hadoop with its tools resolve and tackle big data and resolve the problems faced by users. Further Improvements that can be made to Pig units are Adding the notion of workspace to each text and Remove the boiler plate code appear where there is more than one test method and also adding standalone utility that reads test configuration and generates a test report.

REFERENCES

[1] "Tom White", The Definitive Guide 3rd Edition", Reilly, 12-14,432-447, May.2012
[2] M. R. Palankar, A. Iamnitchi, M. Ripeanu, and S. Garfinkel, "Amazon S3 for science grids: a viable solution?" in DADC '08: Proceedings of the 2008 international workshop on Data-aware distributed computing, 2008, pp. 55–64.
[3] M. Isard et al. Dryad: Distributed data-parallel programs from sequential building blocks. In European Conference on Computer Systems (EuroSys), pages59 {72, Lisbon, Portugal, March 21-23 2007.
[4] R. Pike, S. Dorward, R. Griesemer, and S. Quinlan, "Interpreting the data: Parallel analysis with Sawzall," Scientific Programming, vol. 13, no. 4, 2005.
[5] J. Dean and S. Ghemawat. MapReduce: Simplified data processing on large clusters. In Proc. OSDI, 2004.