

(A unit of Nitte Education Trust)

Nitte - 574 110, Karkala taluk, Udupi Dist., Karnataka

**Department of Computer Science and Engineering**

**BIG DATA ANALYTICS PROJECT**

**ON**

**ANALYSIS AND VISUALIZATION OF ONLINE SHOPPING DATA USING BIG DATA TOOLS**

**PROJECT ASSOCIATES**

**SHIVANI SHENOY B SOWJANYA P**

**4NM17CS166 4NM17CS184**

**VI SEM ‘D’ SECTION VI SEM ‘D’ SECTION**

**DEPT OF CSE DEPT OF CSE**

**PROJECT GUIDES**

**MRS. DIVYA JENIFFER D’SOUZA**

**DEPT OF CSE**

**NMAMIT**

TABLE OF CONTENTS

CONTENTS PAGE NO

1. ABSTRACT 3
2. INTRODUCTION 3-4
3. IMPLEMENTATION
4. TOOLS USED 5
5. EXPECTED RESULT 5
6. OBTAINED RESULTS
7. CONCLUSION
8. REFERENCES 5

**1.ABSTRACT**

The wide spread use of internet and increased web applications accelerate the rampant growth of web content. Every organization produces huge amount of data in different forms like text, audio, video etc., from Multiple sources. The log data stored in web servers is a great source of knowledge. The real challenge for any organization is to understand the behavior of their customers. Analyzing such web log data will help the organizations to understand navigational patterns and interests of their users. As the logs are growing in size day by day, the existing database technologies face a bottleneck to process such massive unstructured data. Hadoop provides a best solution to this problem. Hadoop framework comes up with Hadoop Distributed File System, a reliable distributed storage for data and MapReduce, a distributed parallel processing for executing large volumes of complex data. Hadoop ecosystem constitutes of several other tools like Pig, Hive, Flume, Sqoop etc., for effective analysis of web log data. To write scripts in Map Reduce, one should acquire a good programming knowledge in Java. However Pig, a simple dataflow language can be easily used to analyze such data.

**2.INTRODUCTION**

**PROCESSING DATA WITH PIG**

**Pig Architecture**

Apache Pig simplifies the processing of large data sets. Writing Map Reduce code requires intrinsic knowledge of Java along with sound programming skills.Even after writing the code, extra time is required for assessment of code. Pig supports complex, nested data structures which differentiates it from SQL supporting flatter data structures. It reduces the length of the code by using multi-query approach. The work done by a 200 lines Java program can be replaced by just 10 lines of Pig code. Although Pig scripts are 50% slower in execution compared to MR programs, they still are very effective in increasing

productivity of data engineers and analysts by saving lots of time in writing phase. Pig is made up of two components Pig Latin language and the execution environment to Pig programs. The architecture for Pig is shown in the figure3. The major constituents of Apache Pig are as mentioned:

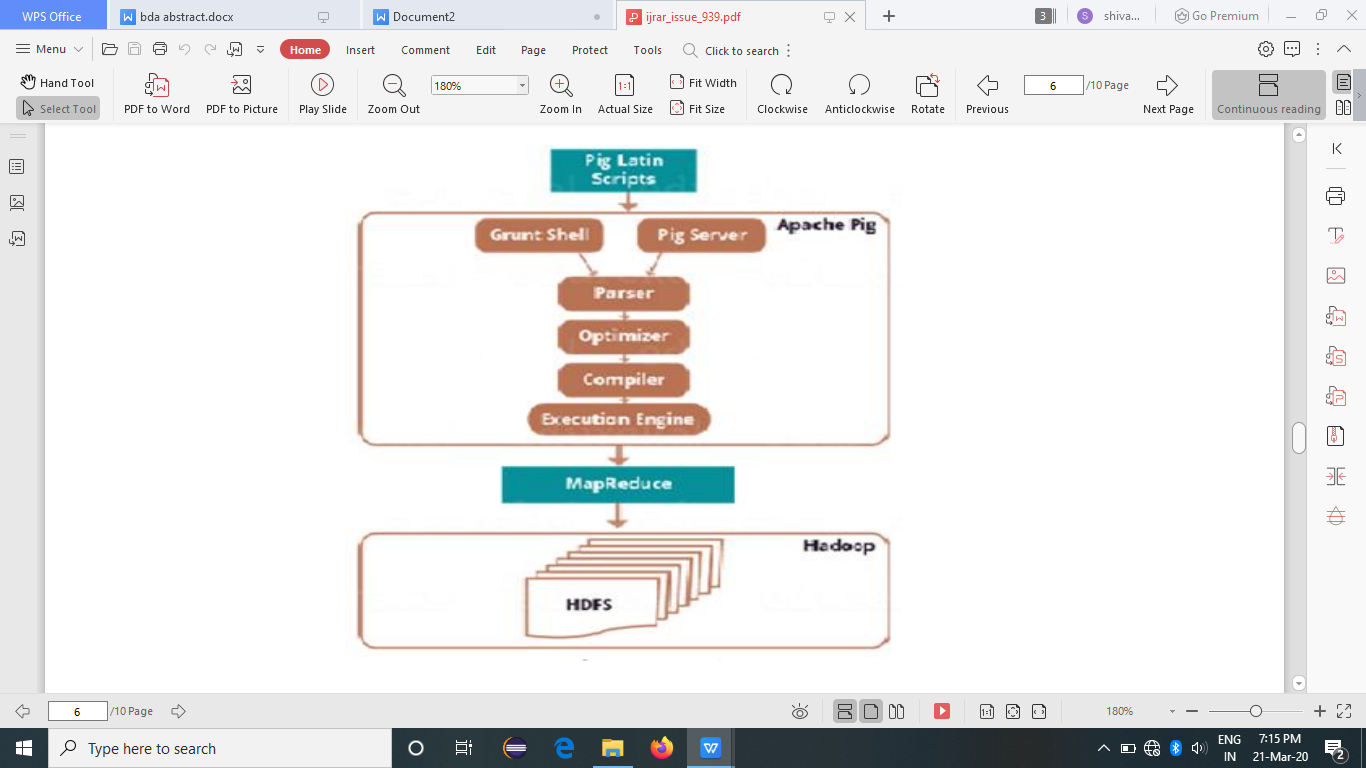


Figure 1. Architecture of Pig

**Parser**: Initially, the Pig Scripts are handled by the Parser. The script is verified for syntax, data types and other miscellaneous details. The output of the parser can be depicted as a DAG (directed acyclic graph). In the DAG, the logical operators of the script are labeled as nodes and the data flows as edges.

**Optimizer**: The logical plan (DAG) is produced for each line of the script after semantic checking and basic parsing. This is sent to the logical optimizer, which performs the logical optimization like projection and pushdown.

**Compiler**: The compiler translates the optimized logical plan into a chain of Map Reduce jobs.

**Execution Engine**: Eventually the execution engine yields the series of Map Reduce jobs to Hadoop in a sorted manner. These Map Reduce jobs are then implemented on HDFS to produce the desired result. Pig has an enriched collection of operators [18] to perform various operations such as sort, filter, join, group, co group etc. Since it is similar to SQL, one can easily write Pig scripts. It provides immense support to specify custom processing in the form of user defined functions (UDFs)

Languages like Java, Python, and JavaScript can be used to implement Pig UDF’s. Pig also facilitates Piggy Bank, a repository for Java UDFs and provides an interface to access Java UDFs written by other users and contribute our own UDFs for use by others. A pig script can be processed in local mode as well as Map Reduce mode.

**Local Mode** : We can choose this mode when we want to parse all input files from the local machine and the local file system. This is very simple and easy to use since there is no involvement of HDFS. This is the best option suitable for initial error detection and correction . To start pig in local mode use,

$ pig –X local

grunt> run ‘/home/User1/pig/query1.pig’

**Map Reduce mode**: The second way of invoking grunt shell is the Map Reduce mode. In order to execute the data residing in HDFS, we need to start Pig in Map Reduce mode. The execution of a Pig script implicitly starts a Map Reduce job in the background to perform specific operations on the input data.

$ pig –X mapreduce hdfs://localhost:9000/pig/Script1.pig

There three ways to implement Pig scripts – Interactive mode, Batch mode and Embedded mode.

**Interactive Mode** (Grunt shell) – The Grunt shell permits to run Apache Pig in interactive mode. In Grunt shell, the Pig Latin statements are executed one by one and eventually the output is triggered using Dump operator. We can use run and exec commands to view the output of a Pig script.

**Batch Mode** (Script) – A number of Pig Latin statements and commands can be grouped and placed in a single file with .pig extension. This file called Pig script, can be run without the intervention of user.

**Embedded Mod**e (UDF) – Since Pig is an efficient data flow language, it allows the user to create his own methods or functions in programming languages like Java, Ruby and Python. These User Defined Functions (UDF’s) can be easily embedded in any pig script.

**3.TOOLS USED**

**1.HADOOP FRAMEWORK**

Large websites that handle millions of simultaneous visitors may generate hundreds of petabytes of logs per day. The traditional RDBMS cannot be used for storage and retrieval of massive heterogeneous datasets.

For efficient and effective analysis of such big weblogs, we need to establish parallel, scalable and faster data mining techniques. Also there is a need for cluster of storage devices and a parallel computing model for loading and examining the data. Hadoop framework satisfies these needs through HDFS (Hadoop Distributed File System) for distributed storage of large datasets and a MapReduce model for parallel processing. Besides these other tools like Sqoop, Flume, Pig, Hive are also available for handling the massive and complex datasets.

**2.APACHE PIG**

Pig an open source high-level data flow system, is a layer of abstraction built on the top of Map Reduce. It can do better on any data size, type or location. Pig has attained popularity since its introduction in 2006 by Yahoo. It promotes an adhoc way of creating and executing Map Reduce jobs on very large data sets. It is better than all RDBMS and DBMS based on performance, storage, and transaction level fault tolerance. Apache Pig has been proven to be the most efficient tool for analyzing structured, semi-structured and unstructured data. Pig uses ETL process for data warehousing. ETL, which stands for “Extract, Transform and Load” is the set of functions combined in one tool to extract large amount of data from numerous databases. Pig furnishes a high-level data flow language called Pig Latin. It has several operators, that help programmers in developing their own functional units for reading input, writing output, and processing the data. Pig queries and functions can be easily translated into a series of Map Reduce jobs which are then run on a Hadoop cluster. Performance, time complexity and the accuracy of data is vital for any enterprise and this can be accomplished through Apache Pig.

**5.EXPECTED RESULT:**

To be able to perform various queries on the large data set and to visualize the various aspects of data present in the online shopping website.

**6.OBTAINED RESULT:**