Session 4: SCHEDULERS IN YARN & INTRODUCTION TO PIG

Assignment 4.3

Student Name: Subbham Vishal

Course: Big Data Hadoop & Spark Training

Assignment 4.3 – Write a program to implement wordcount using Pig.
Share the screenshots of the commands used with its associated output.

Contents

Problem Statement

Write a program to implement word count using Pig. Share the screenshots of the commands used with its associated output.

Introduction

In this assignment we are going to write **word count** program using Pig Latin.

Input Texts

We have the below words in the text file wordcountpig.txt which is placed at /home/acadgild/hadoop.

"Write a program to implement wordcount using Pig. Share the screenshots of the commands used with its associated output.

Here we will write a simple pig script for the word count program."

PIG Commands – Word Count Example

1. Load the data from HDFS

"Word_Count = LOAD '/home/acadgild/hadoop/wordcountpig.txt' USING PigStorage(',') AS (line:chararray);"

```
grunt>
grunt> Word_Count = LOAD '/home/acadgild/hadoop/wordcountpig.txt' USING PigStorage(',') AS (line:chararray);
```

2. Convert the Sentence into words and Convert Column into Rows

"Convert_Words = FOREACH Word_Count GENERATE FLATTEN(TOKENIZE(line,' ')) AS word;"

```
grunt>
grunt> Convert_Words = FOREACH Word_Count GENERATE FLATTEN(TOKENIZE(line,' ')) AS word;
grunt>
```

```
(Write)
(a)
(program)
(to)
(implement)
(wordcount)
(using)
(Pig.Share)
(the)
(screenshots)
(of)
(the)
(commands)
(used)
(with)
(its)
(associated)
(output.)
(Here)
(we)
(will)
(write)
(a)
(simple)
(pig)
(script)
(for)
(the)
(word)
(count)
(program.)
grunt>
grunt>
```

3. Apply GROUP BY

"Grouped = GROUP Convert_Words BY word;"

```
grunt>
grunt> Grouped = GROUP Convert_Words BY word;
grunt>
```

```
(a,{(a),(a)})
(of,{(of)})
(to,{(to)})
(we,{(we)})
(for,{(for)})
(its,{(its)})
(pig,{(pig)})
(the,{(the),(the),(the)})
(Here,{(Here)})
(used,{(used)})
(will,{(will)})
(with,{(with)})
(word,{(word)})
(Write,{(Write)})
(count,{(count)})
(using,{(using)})
(write,{(write)})
(script,{(script)})
(simple,{(simple)})
(output.,{(output.)})
(program,{(program)})
(commands,{(commands)})
(program.,{(program.)})
(Pig.Share,{(Pig.Share)})
(implement,{(implement)})
(wordcount,{(wordcount)})
(associated,{(associated)})
(screenshots,{(screenshots)})
grunt>
```

4. Generate Word Count

"wordcountpig = FOREACH Grouped GENERATE group, COUNT(Convert_Words);

```
(word:chararray)}.
Details at logfile: /home/acadgild/pig_1508393548384.log
grunt> wordcountpig = FOREACH Grouped GENERATE group, COUNT(Convert_Words);
grunt>
```

Expected Output

DUMP wordcountpig;

```
(a,2)
(of,1)
(to,1)
(we,1)
(for,1)
(its,1)
(pig,1)
(the,3)
(Here,1)
(used,1)
(will,1)
(with,1)
(word,1)
(Write,1)
(count,1)
(using,1)
(write,1)
(script,1)
(simple,1)
(output.,1)
(program,1)
(commands,1)
(program.,1)
(Pig.Share,1)
(implement,1)
(wordcount,1)
(associated,1)
(screenshots,1)
```

Complete PIG Code

```
Word_Count = LOAD '/home/acadgild/hadoop/wordcountpig.txt' USING PigStorage(',') AS
(line:chararray);

Convert_Words = FOREACH Word_Count GENERATE FLATTEN(TOKENIZE(line,' ')) AS word;

Grouped = GROUP Convert_Words BY word;

wordcountpig = FOREACH Grouped GENERATE group, COUNT(Convert_Words);

DUMP wordcountpig;
```

xxxxxx----------COMPLETED----------xxxxxx