**PIG MapReduce - Word Count Lab Manual**

**Aim:**

**To implement Word Count using Apache Pig and explore additional commands for efficient data processing in Hadoop.**

**Understanding Apache Pig in MapReduce:**

**Apache Pig translates high-level scripts into MapReduce jobs, simplifying large-scale data handling.**

**1. Loading Data into Pig**

**Command:**

**lines = LOAD '/hdfs\_path/input.txt' USING PigStorage('\n') AS (line:chararray);**

**2. Tokenizing Words**

**Command:**

**words = FOREACH lines GENERATE FLATTEN(TOKENIZE(line)) AS word;**

**3. Grouping Words**

**Command:**

**grouped\_words = GROUP words BY word;**

**4. Counting Words**

**Command:**

**word\_count = FOREACH grouped\_words GENERATE group AS word, COUNT(words) AS count;**

**5. Storing the Word Count Output**

**Command:**

**STORE word\_count INTO '/hdfs\_path/output\_wordcount' USING PigStorage(',');**

**6. Filtering Specific Words (Example: Filtering words with more than 5 letters)**

**Command:**

**filtered\_words = FILTER words BY SIZE(word) > 5;**

**7. Sorting Word Count Results**

**Command:**

**sorted\_word\_count = ORDER word\_count BY count DESC;**

**8. Limiting Results (Example: Top 10 Most Frequent Words)**

**Command:**

**top\_words = LIMIT sorted\_word\_count 10;**

**9. Removing Duplicate Words**

**Command:**

**distinct\_words = DISTINCT words;**

**10. Joining with Another Dataset (Example: Adding Stop Words Dataset for Filtering)**

**Command:**

**stopwords = LOAD '/hdfs\_path/stopwords.txt' USING PigStorage('\n') AS (stopword:chararray);**

**filtered\_word\_list = FILTER words BY NOT (word IN stopwords);**

**11. Splitting Data into Training and Testing Sets**

**Command:**

**SPLIT word\_count INTO training\_data IF count > 100, testing\_data OTHERWISE;**

**12. Storing Processed Data in a Different Format (Example: JSON Storage)**

**Command:**

**STORE sorted\_word\_count INTO '/hdfs\_path/output\_json' USING JsonStorage();**

**Example Output (for Word Count):**

| **Word** | **Count** |
| --- | --- |
| **Hadoop** | **150** |
| **Pig** | **120** |
| **Data** | **90** |
| **Hive** | **80** |

**Final Result:**

**Apache Pig successfully processes text data using MapReduce, performs word count analysis, and allows further data manipulation using various Pig commands. This method simplifies big data analytics, providing an alternative to complex Java-based MapReduce programming.**