Practical 13

Aim:

Locate dataset (e.g. sample_weather.txt) for working on weather data which reads the text input files and finds average for temperature, dew point and wind speed.

Theory:

Introduction to Hadoop and MapReduce:

Hadoop is an open-source framework designed to store and process large-scale datasets efficiently using distributed computing. It provides the ability to process large volumes of data by distributing the workload across multiple nodes in a cluster.

Hadoop consists of four key components:

1. **HDFS (Hadoop Distributed File System):** Stores large datasets across multiple nodes.
2. **MapReduce:** A programming model for parallel data processing.
3. **YARN (Yet Another Resource Negotiator):** Manages computing resources in Hadoop clusters.
4. **Hadoop Common:** Provides utilities to support other Hadoop components.

Understanding MapReduce

MapReduce is a framework used for parallel processing of large-scale data. It follows a two-stage process:

- **Mapper:** Reads input data and transforms it into key-value pairs.
- **Reducer:** Aggregates key-value pairs and computes the final results.

This distributed model enables scalable and efficient data analysis for large datasets such as weather records.

System Requirements:

- **Software:**
  - Java JDK 8 or later
  - Apache Hadoop (Standalone Mode)
  - Eclipse/IntelliJ (optional for development)
- **Hardware:**
  - Minimum 4GB RAM
  - Multi-core processor
  - Sufficient disk space for Hadoop installation
- **Operating System:**

- Windows/Linux/MacOS

Algorithm for Weather Data Processing using MapReduce:

Mapper Phase:

- Read the weather data file line by line.
- Extract temperature, dew point, and wind speed values.
- Emit a key-value pair where the key is a field type (e.g., "temperature") and the value is the extracted numeric data.

Reducer Phase:

- Accept key-value pairs from the Mapper.
- Compute the average for each field (temperature, dew point, wind speed).
- Write the final results as (field type, average value).

Implementation Details:

Step 1: Setting up Hadoop in Standalone Mode

1. Install and configure Hadoop.
2. Set environment variables (HADOOP_HOME, JAVA_HOME).
3. Configure Hadoop files (core-site.xml, hdfs-site.xml, mapred-site.xml).
4. Format HDFS and start Hadoop services.

Step 2: Writing Java Code for Weather Data Processing

1. **Create a Java class named WeatherAnalysis.java.**
2. **Implement the Mapper class to extract weather parameters:**
   - Parse each line of the dataset.
   - Identify and extract values for temperature, dew point, and wind speed.
   - Emit key-value pairs (e.g., <"temperature", 25>).
3. **Implement the Reducer class to calculate averages:**
   - Sum up all values for each weather parameter.
   - Compute the average.
   - Write the final result.
4. **Define the driver class to configure and execute the MapReduce job.**

Step 3: Compiling and Running the Program

1. **Compile the Java code using javac:**

   ```
   javac -classpath `hadoop classpath` -d . WeatherAnalysis.java
   ```

2. **Create a JAR file:**

```
jar -cvf WeatherAnalysis.jar -C . .
```

3. **Run the program using the Hadoop command:**

```
hadoop jar WeatherAnalysis.jar WeatherAnalysis /input /output
```

Code Explanation:

- **Mapper Class:** Extracts temperature, dew point, and wind speed from weather records.
- **Reducer Class:** Aggregates the extracted values and computes their averages.
- **Driver Class:** Configures the Hadoop job, sets input and output paths, and starts execution.

Dataset Format (sample_weather.txt):

A sample weather dataset may contain entries in the following format:

Date	Temp (C)	DewPoint (C)	WindSpeed (km/h)
2024-04-01	25	15	10
2024-04-02	27	16	12
2024-04-03	22	14	8

Expected Output:

```
Temperature Average: 24.67°C
Dew Point Average: 15°C
Wind Speed Average: 10 km/h
```

Advantages of Using Hadoop for Weather Data Processing:

1. **Scalability:** Can process large-scale weather datasets efficiently.
2. **Fault Tolerance:** Ensures reliability by distributing data across multiple nodes.
3. **Parallel Processing:** Executes tasks across multiple nodes to improve performance.
4. **Real-time Insights:** Enables large-scale weather monitoring and forecasting.

Conclusion:

This experiment demonstrated the implementation of a distributed weather data processing application using Hadoop MapReduce. By leveraging parallel computing, Hadoop processes large-scale weather datasets efficiently, providing valuable insights for analysis.