

Ex.No: 9

Date:
6.10.25**Processing Text Input with MapReduce****Aim:**

To process a dataset of daily weather records and find the maximum temperature recorded for each year using MapReduce in Hadoop.

Procedure:**1. Prepare the input text file:**

- Create a plain text file sample.txt with several lines of text.
- Example content:

```
Year, Month, Day, MaxTemperature, MinTemperature
2023,01,01,35,20
2023,01,02,32,18
2024,01,01,36,22
```

○

2. Create Mapper Class:

Reads each line, extracts the year and the maximum temperature, and emits (year, max_temp).

3. Create Reducer Class:

Receives all temperatures for a year and finds the **maximum temperature** for that year.

4. Create Driver Class:

- Configure the job with Mapper, Reducer, InputFormat, OutputFormat, etc.

5. Compile the Java Program:

- Use javac and jar to package the code into a .jar file.

6. Copy input file to HDFS:

```
hdfs dfs -mkdir /input
```

```
hdfs dfs -put sample.txt /input/
```

7. Run the MapReduce job:

```
hadoop jar TextProcessing.jar WordCount /input /output
```

8. Check the output:

```
hdfs dfs -cat /output/part-r-00000
```

Program:**WordCountMapper.java**

```
import java.io.IOException;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Mapper;

public class MaxTempMapper extends Mapper<LongWritable, Text, Text, IntWritable> {
    private Text year = new Text();
    private IntWritable temp = new IntWritable();

    public void map(LongWritable key, Text value, Context context) throws IOException, InterruptedException {
        String line = value.toString();
        String[] fields = line.split(",");
        if (!fields[0].equals("Year")) { // skip header
            year.set(fields[0]);
            temp.set(Integer.parseInt(fields[3]));
            context.write(year, temp);
        }
    }
}
```

WordCountReducer.java

```
import java.io.IOException;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Reducer;

public class MaxTempReducer extends Reducer<Text, IntWritable, Text, IntWritable> {
    private IntWritable result = new IntWritable();
}
```

```
public void reduce(Text key, Iterable<IntWritable> values, Context context) throws IOException,
InterruptedException {
    int maxTemp = Integer.MIN_VALUE;
    for (IntWritable val : values) {
        maxTemp = Math.max(maxTemp, val.get());
    }
    result.set(maxTemp);
    context.write(key, result);
}
```

WordCountDriver.java

```
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

public class MaxTemperature {
    public static void main(String[] args) throws Exception {
        Configuration conf = new Configuration();
        Job job = Job.getInstance(conf, "Max Temperature");
        job.setJarByClass(MaxTemperature.class);
        job.setMapperClass(MaxTempMapper.class);
        job.setCombinerClass(MaxTempReducer.class);
        job.setReducerClass(MaxTempReducer.class);
        job.setOutputKeyClass(Text.class);
        job.setOutputValueClass(IntWritable.class);
        FileInputFormat.addInputPath(job, new Path(args[0]));
        FileOutputFormat.setOutputPath(job, new Path(args[1]));
        System.exit(job.waitForCompletion(true) ? 0 : 1);
    }
}
```

Sample input file (weather.txt)

```
Year,Month,Day,MaxTemperature,MinTemperature
2023,01,01,35,20
2023,01,02,32,18
2023,01,03,37,22
2024,01,01,36,22
2024,01,02,38,21
```

Output:

```
2023    37
2024    38
```

Result:

Thus, the program has been executed successfully and the output is also verified.