Exp No: 11 **Date:**

HADOOP
IMPLEMENT THE MAX TEMPERATURE MAPREDUCE PROGRAM TO IDENTIFY THE YEAR WISE MAXIMUM TEMPERATURE FROM SENSOR DATA

AIM

To implement the Max temperature MapReduce program to identify the year-wise maximum temperature from the sensor data.

Description

Sensors senses weather data in big text format containing station ID, year, date, time, temperature, quality etc. from each sensor and store it in a single line. Suppose thousands of data sensors are there, then we have thousands of records with no particular order. We require only a year and maximum temperature of particular quality in that year.

For example:

Input string from sensor:
002902907099999*1902*010720004+64333+023450
FM-12+
000599999V0202501N027819999999N0000001N9-*00331*+
99999098351ADDGF102991999999999999999999

Here: 1902 is year

0033 is temperature

1 is measurement quality (Range between 0 or 1 or 4 or 5 or 9)

Here each mapper takes the input key as "byte offset of line" and value as "one weather sensor read i.e one line". and parse each line and produce an intermediate key "year" and intermediate value as "temperature of certain measurement qualities" for that year.

The combiner will form set values of temperature. Year and set of values of temperatures is given as input <key, value> to reducer and Reducer will produce year and maximum temperature for that year from the set of temperature values.

PROGRAM

*/

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

//Mapper class


class MaxTemperatureMapper
extends Mapper<LongWritable, Text, Text, IntWritable> { private static final int MISSING

= 9999;

@Override
public void map(LongWritable key, Text value, Context context) throws IOException,
InterruptedException {

String line = value.toString(); String year = line.substring(15, 19); int airTemperature;
if (line.charAt(87) == '+') { // parseInt doesn't like leading plus signs airTemperature =
Integer.parseInt(line.substring(88, 92));
} else {
airTemperature = Integer.parseInt(line.substring(87, 92));
}
String quality = line.substring(92, 93);
if (airTemperature != MISSING && quality.matches("[01459]")) { context.write(new
Text(year), new IntWritable(airTemperature));
}
}
}

//Reducer class
class MaxTemperatureReducer
extends Reducer<Text, IntWritable, Text, IntWritable> {

@Override
public void reduce(Text key, Iterable<IntWritable> values, Context context)
throws IOException, InterruptedException {
```

```
int maxValue = Integer.MIN_VALUE; for (IntWritable value : values) {
maxValue = Math.max(maxValue, value.get());
}
context.write(key, new IntWritable(maxValue));
}
}
//Driver Class

public class MaxTemperature {

public static void main(String[] args) throws Exception { if (args.length != 2) {
System.err.println("Usage: MaxTemperature <input path=""> <output path>"); System.exit(-
1);
}

Job job = Job.getInstance(new Configuration()); job.setJarByClass(MaxTemperature.class);
job.setJobName("Max temperature");

FileInputFormat.addInputPath(job, new Path(args[0])); FileOutputFormat.setOutputPath(job,
new Path(args[1]));


job.setMapperClass(MaxTemperatureMapper.class);
job.setReducerClass(MaxTemperatureReducer.class);

job.setOutputKeyClass(Text.class); job.setOutputValueClass(IntWritable.class);

job.submit();
}
}
```

OUTPUT:

Input for String :

0029029070999991902010720004+64333+023450FM-12+
000599999V0202501N027819999999N0000001N9-00331+
99999098351ADDGF10299199999999999999’

```
harithaah@vbox:~$ start-all.sh
WARNING: Attempting to start all Apache Hadoop daemons as harithaah in 10 seconds.
WARNING: This is not a recommended production deployment configuration.
WARNING: Use CTRL-C to abort.
Starting namenodes on [localhost]
Starting datanodes
Starting secondary namenodes [vbox]
vbox: Warning: Permanently added 'vbox' (ED25519) to the list of known hosts.
Starting resourcemanager
Starting nodemanagers
```

```
harithaah@vbox:~$ jps
5746 DataNode
5572 NameNode
6009 SecondaryNameNode
6955 Jps
6316 ResourceManager
6509 NodeManager
```

```
harithaah@vbox:~/CC/exp3$ javac -classpath /home/harithaah/hadoop/share/hadoop/common/*:/home/harithaah/hadoop/share/hadoop/common/lib/*:/home/harithaah/hadoop/share/hadoop/mapreduce/*:/home/harithaah/hadoop/share/hadoop/hdfs/* -d . Maxtemperaturemapper.java Maxtemperaturereducer.java Maxtemperature.java
harithaah@vbox:~/CC/exp3$ ls
Maxtemperature.class  MaxTemperature.java         Maxtemperaturemapper.java    Maxtemperaturereducer.java
Maxtemperature.java   Maxtemperaturemapper.class  Maxtemperaturereducer.class
harithaah@vbox:~/CC/exp3$ ls
Maxtemperature.class  MaxTemperature.java         Maxtemperaturemapper.java    Maxtemperaturereducer.java
Maxtemperature.java   Maxtemperaturemapper.class  Maxtemperaturereducer.class
```

```
harithaah@vbox:~/CC/exp3$ hdfs dfs -mkdir /temperature
harithaah@vbox:~/CC/exp3$ hdfs dfs -put data.txt /temperature
harithaah@vbox:~/CC/exp3$ hdfs dfs -ls /temperature
Found 1 items
-rw-r--r--   1 harithaah supergroup        428 2024-11-16 20:49 /temperature/data.txt
```

```
harithaah@vbox:~/CC/exp3$ jar -cvf Maxtemperature.jar -C . .
added manifest
adding: Maxtemperaturemapper.java(in = 984) (out= 425)(deflated 56%)
adding: Maxtemperaturereducer.java(in = 590) (out= 289)(deflated 51%)
adding: Maxtemperature.java(in = 1122) (out= 440)(deflated 60%)
adding: Maxtemperaturemapper.class(in = 1876) (out= 806)(deflated 57%)
adding: Maxtemperaturereducer.class(in = 1687) (out= 717)(deflated 57%)
adding: Maxtemperature.class(in = 1558) (out= 863)(deflated 44%)
adding: data.txt(in = 428) (out= 102)(deflated 76%)
```

```
harithaah@vbox:~/CC/exp3$ hadoop jar Maxtemperature.jar Maxtemperature /temperature/data.txt /temperature/output
2024-11-16 21:03:46,966 INFO client.DefaultNoHARMFailoverProxyProvider: Connecting to ResourceManager at /0.0.0.0:8032
2024-11-16 21:03:48,031 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and execute your application with ToolRunner to remedy this.
2024-11-16 21:03:48,166 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/staging/harithaah/.staging/job_1731768888369_0002
2024-11-16 21:03:49,221 INFO input.FileInputFormat: Total input files to process : 1
2024-11-16 21:03:50,247 INFO mapreduce.JobSubmitter: number of splits:1
2024-11-16 21:03:50,901 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1731768888369_0002
2024-11-16 21:03:50,901 INFO mapreduce.JobSubmitter: Executing with tokens: []
2024-11-16 21:03:51,277 INFO conf.Configuration: resource-types.xml not found
2024-11-16 21:03:51,278 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
2024-11-16 21:03:51,452 INFO impl.YarnClientImpl: Submitted application application_1731768888369_0002
2024-11-16 21:03:51,572 INFO mapreduce.Job: The url to track the job: http://vbox:8088/proxy/application_1731768888369_0002/
2024-11-16 21:03:51,574 INFO mapreduce.Job: Running job: job_1731768888369_0002
2024-11-16 21:04:05,096 INFO mapreduce.Job: Job job_1731768888369_0002 running in uber mode : false
2024-11-16 21:04:05,097 INFO mapreduce.Job:  map 0% reduce 0%
2024-11-16 21:04:14,255 INFO mapreduce.Job:  map 100% reduce 0%
2024-11-16 21:04:22,408 INFO mapreduce.Job:  map 100% reduce 100%
2024-11-16 21:04:24,465 INFO mapreduce.Job: Job job_1731768888369_0002 completed successfully
2024-11-16 21:04:24,642 INFO mapreduce.Job: Counters: 54
        File System Counters
                FILE: Number of bytes read=6
                FILE: Number of bytes written=617801
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=535
                HDFS: Number of bytes written=0
                HDFS: Number of read operations=8
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=2
```

```
		Map output bytes=0
		Map output materialized bytes=6
		Input split bytes=107
		Combine input records=0
		Combine output records=0
		Reduce input groups=0
		Reduce shuffle bytes=6
		Reduce input records=0
		Reduce output records=0
		Spilled Records=0
		Shuffled Maps =1
		Failed Shuffles=0
		Merged Map outputs=1
		GC time elapsed (ms)=222
		CPU time spent (ms)=3890
		Physical memory (bytes) snapshot=551587840
		Virtual memory (bytes) snapshot=5162274816
		Total committed heap usage (bytes)=427819008
		Peak Map Physical memory (bytes)=328470528
		Peak Map Virtual memory (bytes)=2577395712
		Peak Reduce Physical memory (bytes)=223117312
		Peak Reduce Virtual memory (bytes)=2584879104
	Shuffle Errors
		BAD_ID=0
		CONNECTION=0
		IO_ERROR=0
		WRONG_LENGTH=0
		WRONG_MAP=0
		WRONG_REDUCE=0
	File Input Format Counters
		Bytes Read=428
	File Output Format Counters
```

RESULT

Thus a java program has been implemented to identify the year-wise maximum temperature from the sensor data.