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(MaxTemperatureMa
pper.class);
job.setReducerClass(MaxTemperatureRe
ducer.class);

job.setOutputKeyClass(Text.class);

job.setOutputValueClass(IntWritable.class);job.submit();
}
}
```

OUTPUT:

Input for String :

002902907099999*1902*010720004+64333+023

450FM-12+
000599999V0202501N027819999999N0000001N9-*00331*+
99999098351ADDGF10299199999999999999’

```
lksh@fedora:~$ start-all.sh
WARNING: Attempting to start all Apache Hadoop daemons as lksh 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 [fedora]
Starting resourcemanager
Starting nodemanagers
lksh@fedora:~$ jps
4643 Jps
3529 DataNode
3386 NameNode
3738 SecondaryNameNode
4108 NodeManager
3951 ResourceManager
lksh@fedora:~$
```

```
lksh@fedora:~/exp3$ nano dataset.txt
lksh@fedora:~/exp3$ hdfs dfs -mkdir /exp2
```

```
lksh@fedora:~/exp3$ hadoop jar $HADOOP_STREAMING -input /exp2/dataset.txt -output /exp2/output1 -mapper ~/exp3/mapper.py -reducer ~/exp3/reducer.py
packageJobJar: [/tmp/hadoop-unjar2773513365584043905/] [] /tmp/streamjob3053124438108899539.jar tmpDir=null
2024-10-12 11:26:24,211 INFO client.DefaultNoHARMFailoverProxyProvider: Connecting to ResourceManager at /0.0.0.0:8032
2024-10-12 11:26:24,695 INFO client.DefaultNoHARMFailoverProxyProvider: Connecting to ResourceManager at /0.0.0.0:8032
2024-10-12 11:26:31,634 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/staging/lksh/.staging/job_1728710244759_0001
2024-10-12 11:26:32,802 INFO mapreduce.JobSubmitter: Cleaning up the staging area /tmp/hadoop-yarn/staging/lksh/.staging/job_1728710244759_0001
2024-10-12 11:26:32,875 ERROR streaming.StreamJob: Error Launching job : Input path does not exist: hdfs://localhost:9000/exp2/dataset.txt
```

```
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
```

```
lksh@fedora:~$ hdfs dfs -cat /exp3/output/part-00000
01      26.5
02      26.6
03      29.1
04      30.8
05      31.1
06      33.6
07      38.5
08      40.2
09      36.5
10      36.9
11      27.6
12      25.9
lksh@fedora:~$
```


RESULT:

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