

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 sense 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:

0029029070999991902010720004+64333+023450

FM-12+

000599999V0202501N0278199999999N0000001N9-00331+

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;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Mapper;
import org.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+
000599999V0202501N0278199999999N0000001N9-00331+
99999098351ADDGF102991999999999999999'

```

```
helen@fedora:~  
helen@fedora:~/exp2  
helen@fedora:~/exp2$ cd  
helen@fedora:~$ start-all.sh  
WARNING: Attempting to start all Apache Hadoop daemons as helen in 10 seconds.  
WARNING: This is not a recommended production deployment configuration.  
WARNING: Use CTRL-C to abort.  
Starting namenodes on [localhost]  
localhost: namenode is running as process 3449. Stop it first and ensure /tmp/hadoop-helen-namenode.pid file is empty before retry.  
Starting datanodes  
localhost: datanode is running as process 3638. Stop it first and ensure /tmp/hadoop-helen-datanode.pid file is empty before retry.  
Starting secondary namenodes [fedora]  
fedora: secondarynamenode is running as process 3924. Stop it first and ensure /tmp/hadoop-helen-secondarynamenode.pid file is empty before retry.  
Starting resourcemanager  
resourcemanager is running as process 4213. Stop it first and ensure /tmp/hadoop-helen-resourcemanager.pid file is empty before retry.  
Starting nodemanagers  
localhost: nodemanager is running as process 4348. Stop it first and ensure /tmp/hadoop-helen-nodemanager.pid file is empty before retry.  
helen@fedora:~$ jps  
3924 SecondaryNameNode  
4213 ResourceManager  
3638 DataNode  
3449 NameNode  
4348 NodeManager  
8061 Jps  
helen@fedora:~$
```

```
helen@fedora:~  
helen@fedora:~/exp2  
helen@fedora:~$ hadoop jar $HADOOP_STREAMING -input /exp2/data.txt -output /exp2/output3 -mapper ~/exp2/mapper.py -reducer ~/exp2/reducer.py  
packageJobJar: [/tmp/hadoop-unjar6285464154385550498/] [] /tmp/streamjob1479607642965193751.jar tmpDir=null  
2024-10-12 07:29:18,538 INFO client.DefaultNoHARMFailoverProxyProvider: Connecting to ResourceManager at /0.0.0.0:8032  
2024-10-12 07:29:18,750 INFO client.DefaultNoHARMFailoverProxyProvider: Connecting to ResourceManager at /0.0.0.0:8032  
2024-10-12 07:29:19,201 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/staging/helen/.staging/job_1728729630364_0002  
2024-10-12 07:29:19,717 INFO mapred.FileInputFormat: Total input files to process : 1  
2024-10-12 07:29:19,994 INFO mapreduce.JobSubmitter: number of splits:2  
2024-10-12 07:29:20,746 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1728729630364_0002  
2024-10-12 07:29:20,746 INFO mapreduce.JobSubmitter: Executing with tokens: []  
2024-10-12 07:29:21,006 INFO conf.Configuration: resource-types.xml not found  
2024-10-12 07:29:21,006 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.  
2024-10-12 07:29:21,187 INFO impl.YarnClientImpl: Submitted application application_1728729630364_0002  
2024-10-12 07:29:21,216 INFO mapreduce.Job: The url to track the job: http://fedora:8088/proxy/application_1728729630364_0002/  
2024-10-12 07:29:21,218 INFO mapreduce.Job: Running job: job_1728729630364_0002  
2024-10-12 07:29:29,626 INFO mapreduce.Job: Job job_1728729630364_0002 running in uber mode : false  
2024-10-12 07:29:29,627 INFO mapreduce.Job: map 0% reduce 0%  
2024-10-12 07:29:39,796 INFO mapreduce.Job: map 100% reduce 0%  
2024-10-12 07:29:46,056 INFO mapreduce.Job: map 100% reduce 100%  
helen@fedora:~$ hdfs dfs -cat /exp2/output3/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  
helen@fedora:~$
```

RESULT

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

Sample Questions

BASIC UNDERSTANDING: Exp 1

1. What is virtualization?

Ans. Virtualization is an abstraction layer that decouples physical hardware from operating system to deliver greater IT resource utilization and flexibility.

2. What is the Difference between Full Virtualization and Para Virtualization?

Ans. Full virtualization & Para virtualization both comes under the Hardware virtualization. Some of the differences between them are listed below:

Full Virtualization: In full virtualization guest VMs (Virtual Machines) are not aware that they are in virtualized environment there-fore the guest os issues command to what it thinks as actual hardware but actually are just simulated devices created by the hosts.

Para Virtualization : In para virtualization the guest vm is aware that it is in a virtualized environment . If guest vm requires resources , it issues commands to host operating system instead of directly communicating with simulated hardware.

3.What is Hyper-visor ?

A **hypervisor** or virtual machine monitor (VMM) is computer software, firmware or hardware that creates and runs virtual machines. A computer on which a **hypervisor** runs one or more virtual machines is called a host machine, and each virtual machine is called a guest machine.

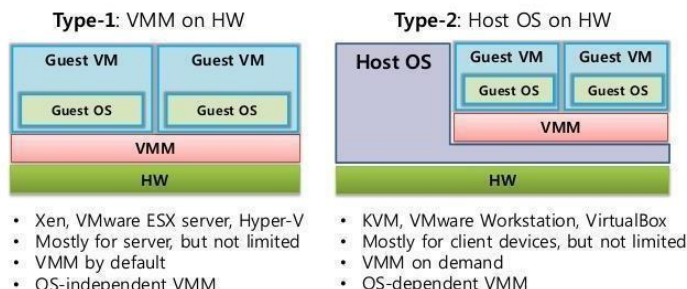
4.What are the difference between Type 1 and Type 2 Hypervisor ?

Ans. Type 1: When the Hypervisor is installed on bare metal / Physical hardware it is known as Type 1 Hypervisor . Examples are VM ware ESXi, Oracle VM, Microsoft Hyper V.

Type 2: When the Hypervisor is installed on top of an operating system it is known as Type 2 Hypervisor . Examples are Microsoft Virtual Server, VM Ware Server and workstation.

Type-1 vs. Type-2

- Depending on what sits right on HW



BASIC UNDERSTANDING: Exp 2

1. What is a virtual block?

A virtual block device is an interface with applications that appears to the applications as a memory device, such as a standard block device.

2. What is a virtual disk?

Virtual disks are stored as files on the host computer or on a network file server. It does not matter whether the physical disk that holds the files is IDE or SCSI.

IDE (Integrated Drive Electronics) SCSI (Small Computer System Interface) SATA (Serial Advanced Technology Attachment)

3. What is a VM clone?

A clone is a copy of an existing virtual machine.

4. What is a Snapshot and a Template?

A snapshot is a copy of the virtual machine's disk file at a given point in time.

Snapshots provide a change log for the virtual disk and are used to restore a VM to a particular point in time when a failure or system error occurs.

A **template** is a master copy of a virtual machine that can be used to create many clones.