BIG DATA ANALYTICS LAB

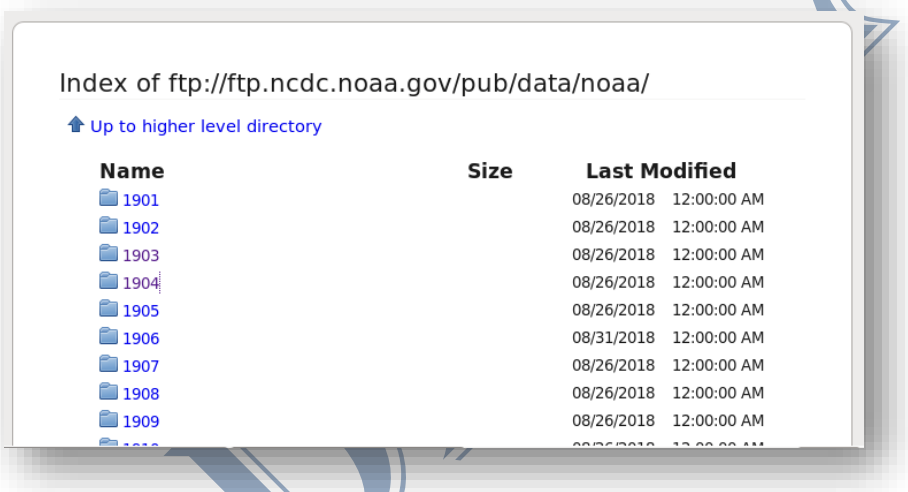
EXPERIMENT_NO-04

AIM: Write a Map Reduce program that mines weather data. Weather sensors collecting data every hour at many locations across the globe gather a large volume of log data, which is a good candidate for analysis with MapReduce, since it is semi structured and record-oriented.

Description:

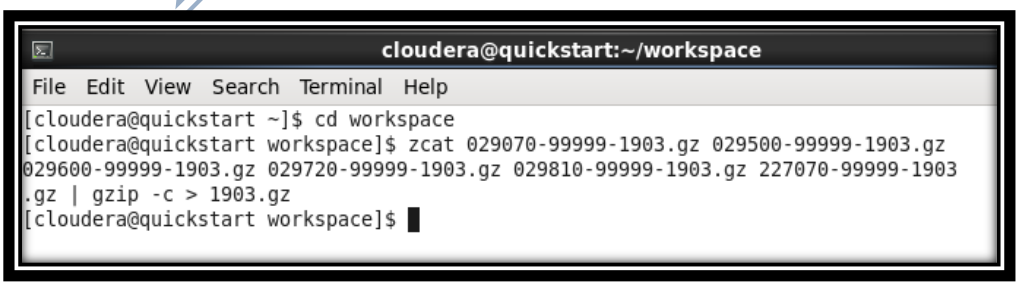
1) Open Oracle VM VirtualBox->export cloudera->start

2) Open browser and type "**ftp://ftp.ncdc.noaa.gov/pub/data/noaa/**".



3) Download any 3 folders to workspace.

4) There are multiple files in a folder, concatenate those files into a folder as follows.



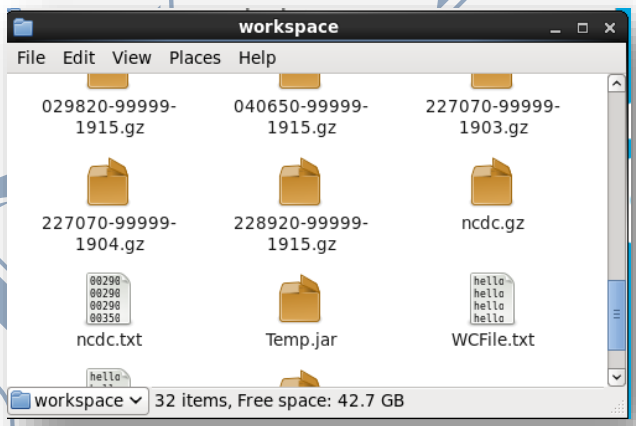
```
[cloudera@quickstart workspace]$ zcat 029070-99999-1904.gz 029500-99999-1904.gz 029600-99999-1904
.gz 029720-99999-1904.gz 029810-99999-1904.gz 227070-99999-1904.gz | gzip -c > 1904.gz
[cloudera@quickstart workspace]$
```

```
[cloudera@quickstart workspace]$ zcat 028060-99999-1915.gz 028690-99999
-1915.gz 028750-99999-1915.gz 029170-99999-1915.gz 029440-99999-1915.gz
  029820-99999-1915.gz 040650-99999-1915.gz 228920-99999-1915.gz | gzip
 -c > 1915.gz
[cloudera@quickstart workspace]$
```

5) Concatenate these 3 folders into a single folder as follows.

```
[cloudera@quickstart workspace]$ zcat 1903.gz 1904.gz 1915.gz | gzip -c > ncdc.gz
[cloudera@quickstart workspace]$
```

6) Right click on ncdc.gz ->Extract here->rename->ncdc.txt



7) In eclipse->File->New->Java project->Project name “Weather”->Finish

8) Create three classes.

Right click on Weather-> New->class->Name “MaxTemperature”

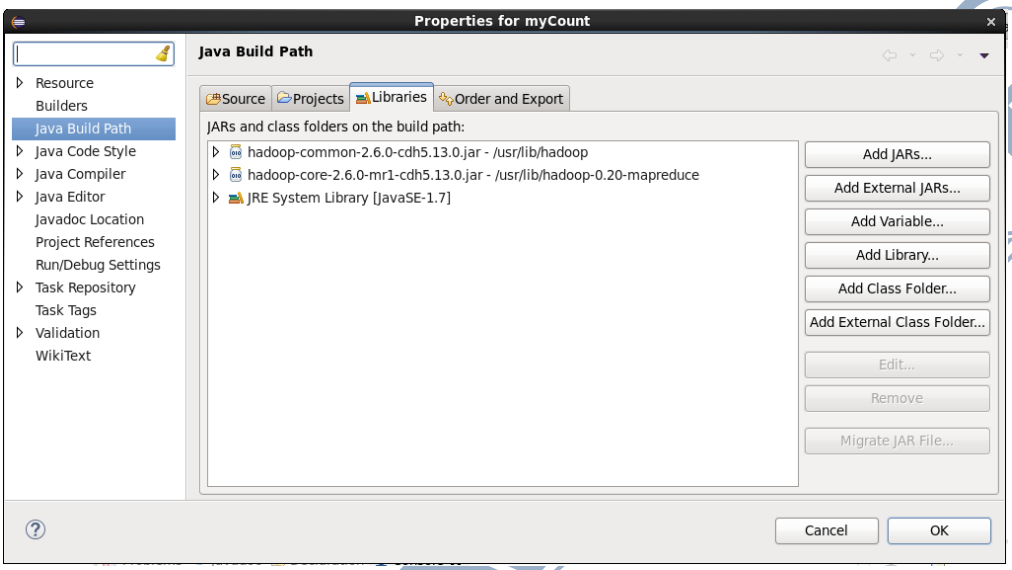
Right click on Weather-> New->class->Name “MaxTemperatureMapper”

Right click on Weather -> New->class->Name “MaxTemperatureReducer”

9) Add Hadoop libraries.

Right click on Weather->Build path->Configure Build path->Add external JARS. (usr\lib\hadoop\hadoop-common-2.6.0-cdh 5.13.0 jar,

usr\lib\hadoop\hadoop-core-2.6.0-cdh 5.13.0 jar)



PROGRAM:

MaxTemperature.java

```
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{
```

```
    if(args.length!=2)
    {
       System.err.println("error");
       System.exit(-1);
    }
    Job job=new Job();
    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);
    System.exit(job.waitForCompletion(true)?0:1);
}
}
```

MaxTemperatureMapper.java

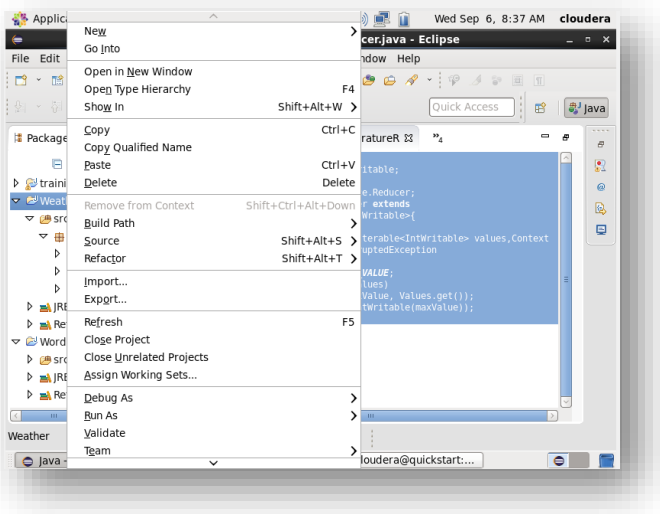
```
import java.io.IOException;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Mapper;
```

```
public class 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)=='+')
          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));
   }
}
```

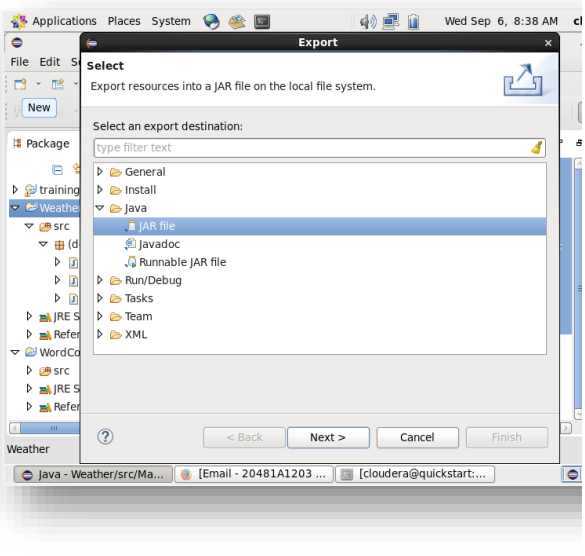
MaxTemperatureReducer.java

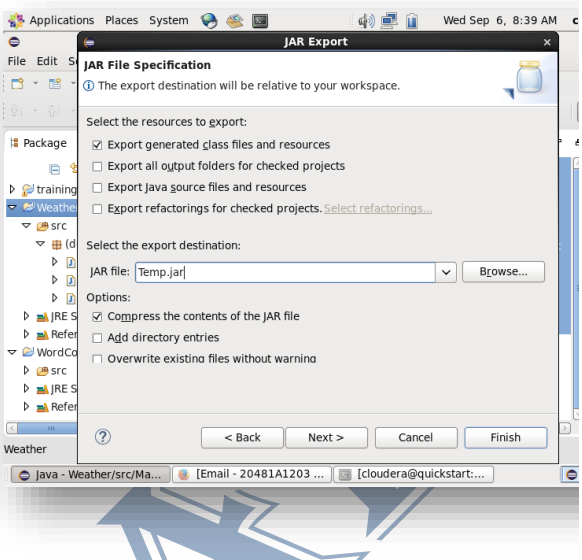
```
import java.io.IOException;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
```

```
import org.apache.hadoop.mapreduce.Reducer;
public class 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 Values:values)
         maxValue=Math.max(maxValue, Values.get());
      context.write(key, new IntWritable(maxValue));
   }
}
```



Right click on Weather->Export->java->jar file->JAR file:" **Temp**"->Finish





OUTPUT:

In Terminal

```
[cloudera@quickstart workspace]$ hadoop fs -put ncdc.txt ncdc.txt
[cloudera@quickstart workspace]$
```

```
[cloudera@quickstart workspace]$ hadoop jar Temp.jar MaxTemperature ncdc.txt out
```

```
[cloudera@quickstart workspace]$ hadoop fs -cat out/part-r-00000
1903    289
1904    256
1915    294
[cloudera@quickstart workspace]$
```