**✅ 4. Common HDFS Commands**

| **Action** | **Command** |
| --- | --- |
| List files | hdfs dfs -ls / |
| Make directory | hdfs dfs -mkdir /user/hduser/input |
| Copy file to HDFS | hdfs dfs -put localfile.txt /user/hduser/input/ |
| Copy file from HDFS | hdfs dfs -get /user/hduser/output/part-00000 ./ |
| Remove file | hdfs dfs -rm /user/hduser/input/localfile.txt |
| Remove directory | hdfs dfs -rm -r /user/hduser/input/ |
| View file contents | hdfs dfs -cat /user/hduser/output/part-00000 |
| Check disk usage | hdfs dfs -du -h / |
| Show filesystem stats | hdfs dfsadmin -report |

**✅ 5. Starting Hadoop Services (if not running)**

If you're running Hadoop in pseudo-distributed mode:

bash

start-dfs.sh start-yarn.sh

You might need to run these with sudo depending on your setup.

**✅ 6. Example Usage**

bash

cd ~/hadoop-3.x.x/ hdfs dfs -mkdir /input hdfs dfs -put ~/myfile.txt /input/ hdfs dfs -ls /input

Would you like help checking if Hadoop is set up correctly on your system?

**You said:**

**Explanation- Lab Programs.pptx (1).ppt**

Presentation

give codes or solutions

**ChatGPT said:**

Based on your request and the contents of the provided PPT, here is a **complete set of Hadoop HDFS terminal commands and code snippets** to run MapReduce jobs in **Ubuntu**.

**✅ HDFS Commands in Terminal (Ubuntu)**

**➤ Start Hadoop (pseudo-distributed mode)**

bash

start-dfs.sh start-yarn.sh

**➤ Copy input file to HDFS**

bash

hdfs dfs -mkdir -p /rgs hdfs dfs -copyFromLocal /home/hduser/Desktop/temp.txt /rgs/avtemp.txt

**➤ Run MapReduce Job for Average Yearly Temperature**

bash

hadoop jar /home/hduser/Downloads/avgtemp.jar temp.AverageDriver /rgs/avtemp.txt /out7 hdfs dfs -cat /out7/\*

Expected output:

python-repl

1901 46

1902 48

...

**➤ Run MapReduce Job for Monthly Mean Max Temperature**

bash

hadoop jar /home/hduser/Downloads/meanmax.jar meanmax.MeanMaxDriver /rgs/avtemp.txt /out8 hdfs dfs -cat /out8/\*

Expected output (example):

python-repl

CopyEdit

01 4

02 0

03 7

...

12 3

**✅ Code Snippet Summary**

**🔹 Average Temperature MapReduce (per year)**

**Driver**: AverageDriver.java  
**Mapper**: AverageMapper.java  
**Reducer**: AverageReducer.java

**🔹 Mean Max Temperature (per month)**

**Driver**: MeanMaxDriver.java  
**Mapper**: MeanMaxMapper.java  
**Reducer**: MeanMaxReducer.java

Already provided in your uploaded file — let me know if you want me to create .java files or help compile them.

**🛠️ How to Compile and Run MapReduce Code**

**➤ Compile Java Code**

bash

cd /home/hduser/Desktop/ javac -classpath `hadoop classpath` -d avgtemp\_classes AverageMapper.java AverageReducer.java AverageDriver.java jar -cvf avgtemp.jar -C avgtemp\_classes/ .

Repeat for meanmax:

bash

CopyEdit

javac -classpath `hadoop classpath` -d meanmax\_classes MeanMaxMapper.java MeanMaxReducer.java MeanMaxDriver.java jar -cvf meanmax.jar -C meanmax\_classes/ .

Would you like me to generate .java source files for both programs or help you run the top 10 words frequency example mentioned in your PPT?

**You said:**

yes

**ChatGPT said:**

Updated saved memory

Great! Below are the complete .java files for the two MapReduce programs based on your PPT:

**✅ 1. Average Yearly Temperature MapReduce**

**🔹**AverageMapper.java

java

CopyEdit

package temp; import java.io.IOException; import org.apache.hadoop.io.IntWritable; import org.apache.hadoop.io.Text; import org.apache.hadoop.mapreduce.Mapper; public class AverageMapper extends Mapper<Object, Text, Text, IntWritable> { private Text year = new Text(); private IntWritable temp = new IntWritable(); public void map(Object key, Text value, Context context) throws IOException, InterruptedException { String[] parts = value.toString().split("\\s+"); if (parts.length == 2) { year.set(parts[0].substring(0, 4)); // Extract year from YYYYMM temp.set(Integer.parseInt(parts[1])); context.write(year, temp); } } }

**🔹**AverageReducer.java

java

CopyEdit

package temp; import java.io.IOException; import org.apache.hadoop.io.IntWritable; import org.apache.hadoop.io.Text; import org.apache.hadoop.mapreduce.Reducer; public class AverageReducer extends Reducer<Text, IntWritable, Text, IntWritable> { public void reduce(Text key, Iterable<IntWritable> values, Context context) throws IOException, InterruptedException { int sum = 0, count = 0; for (IntWritable val : values) { sum += val.get(); count++; } context.write(key, new IntWritable(sum / count)); } }

**🔹**AverageDriver.java

java

CopyEdit

package temp; 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 AverageDriver { public static void main(String[] args) throws Exception { Configuration conf = new Configuration(); Job job = Job.getInstance(conf, "Average Temperature"); job.setJarByClass(AverageDriver.class); job.setMapperClass(AverageMapper.class); job.setReducerClass(AverageReducer.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); } }

**✅ 2. Mean Max Temperature per Month MapReduce**

**🔹**MeanMaxMapper.java

java

CopyEdit

package meanmax; import java.io.IOException; import org.apache.hadoop.io.IntWritable; import org.apache.hadoop.io.Text; import org.apache.hadoop.mapreduce.Mapper; public class MeanMaxMapper extends Mapper<Object, Text, Text, IntWritable> { private Text month = new Text(); private IntWritable temp = new IntWritable(); public void map(Object key, Text value, Context context) throws IOException, InterruptedException { String[] parts = value.toString().split("\\s+"); if (parts.length == 2) { month.set(parts[0].substring(4, 6)); // Extract MM from YYYYMM temp.set(Integer.parseInt(parts[1])); context.write(month, temp); } } }

**🔹**MeanMaxReducer.java

java

CopyEdit

package meanmax; import java.io.IOException; import org.apache.hadoop.io.IntWritable; import org.apache.hadoop.io.Text; import org.apache.hadoop.mapreduce.Reducer; public class MeanMaxReducer extends Reducer<Text, IntWritable, Text, IntWritable> { public void reduce(Text key, Iterable<IntWritable> values, Context context) throws IOException, InterruptedException { int sum = 0, count = 0; for (IntWritable val : values) { sum += val.get(); count++; } context.write(key, new IntWritable(sum / count)); } }

**🔹**MeanMaxDriver.java

java

CopyEdit

package meanmax; 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 MeanMaxDriver { public static void main(String[] args) throws Exception { Configuration conf = new Configuration(); Job job = Job.getInstance(conf, "Mean Max Temp"); job.setJarByClass(MeanMaxDriver.class); job.setMapperClass(MeanMaxMapper.class); job.setReducerClass(MeanMaxReducer.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); } }

Would you like me to give you the commands to compile and run these on your system, or package them as .jar files?

**You said:**

how to run this cd /home/hduser/Desktop/ javac -classpath hadoop classpath -d avgtemp\_classes AverageMapper.java AverageReducer.java AverageDriver.java jar -cvf avgtemp.jar -C avgtemp\_classes/ .

**ChatGPT said:**

Here's a **step-by-step guide** to running your MapReduce program using those commands:

**✅ Step-by-Step to Compile and Package**avgtemp**MapReduce Program**

**✅ 1. Save Java Files**

Save the following 3 files in the folder /home/hduser/Desktop/ with exact names:

* AverageMapper.java
* AverageReducer.java
* AverageDriver.java

Make sure they are in the package temp, so your folder structure should be like:

bash

CopyEdit

/home/hduser/Desktop/temp/AverageMapper.java

/home/hduser/Desktop/temp/AverageReducer.java

/home/hduser/Desktop/temp/AverageDriver.java

You can create the temp directory and move your files:

bash

CopyEdit

cd /home/hduser/Desktop/ mkdir temp # Move your Java files to this folder mv Average\*.java temp/

**✅ 2. Compile Java Files**

Now compile them with Hadoop's classpath:

bash

CopyEdit

cd /home/hduser/Desktop/ javac -classpath `hadoop classpath` -d avgtemp\_classes temp/AverageMapper.java temp/AverageReducer.java temp/AverageDriver.java

This command:

* Uses Hadoop’s libraries for compilation.
* Outputs .class files to a directory named avgtemp\_classes.

**✅ 3. Create JAR File**

Now create a JAR file from the compiled class files:

bash

CopyEdit

jar -cvf avgtemp.jar -C avgtemp\_classes/ .

This packages the classes into avgtemp.jar, which you’ll use to run the MapReduce job.

**✅ 4. Copy Input File to HDFS**

If you haven't already:

bash

CopyEdit

hdfs dfs -mkdir -p /rgs hdfs dfs -copyFromLocal /home/hduser/Desktop/temp.txt /rgs/avtemp.txt

**✅ 5. Run the Job**

bash

CopyEdit

hadoop jar avgtemp.jar temp.AverageDriver /rgs/avtemp.txt /out7

**✅ 6. View Output**

bash

CopyEdit

hdfs dfs -cat /out7/\*

Would you like me to do the same setup for the meanmax program?

**You said:**

yes

**ChatGPT said:**

Great! Here’s how to **compile and run the**meanmax**MapReduce program** step-by-step:

**✅ Step-by-Step: Compile & Run**meanmax**Program**

**✅ 1. Save Java Files**

Save these 3 files with exact names inside a folder named meanmax in your Desktop directory:

* MeanMaxMapper.java
* MeanMaxReducer.java
* MeanMaxDriver.java

So your directory structure will be:

bash

CopyEdit

/home/hduser/Desktop/meanmax/MeanMaxMapper.java

/home/hduser/Desktop/meanmax/MeanMaxReducer.java

/home/hduser/Desktop/meanmax/MeanMaxDriver.java

To organize them properly:

bash

CopyEdit

cd /home/hduser/Desktop/ mkdir meanmax # Move Java files into the meanmax folder mv MeanMax\*.java meanmax/

**✅ 2. Compile Java Files**

Run this to compile all three Java files and place the .class files into meanmax\_classes:

bash

CopyEdit

cd /home/hduser/Desktop/ javac -classpath `hadoop classpath` -d meanmax\_classes meanmax/MeanMaxMapper.java meanmax/MeanMaxReducer.java meanmax/MeanMaxDriver.java

**✅ 3. Create JAR File**

Package the compiled .class files into meanmax.jar:

bash

CopyEdit

jar -cvf meanmax.jar -C meanmax\_classes/ .

**✅ 4. Run the MapReduce Job**

Assuming your input file avtemp.txt is already in HDFS (/rgs/avtemp.txt):

bash

CopyEdit

hadoop jar meanmax.jar meanmax.MeanMaxDriver /rgs/avtemp.txt /out8

**✅ 5. View the Output**

bash

CopyEdit

hdfs dfs -cat /out8/\*

This should display the average temperature per month (01 to 12).

Let me know if you’d like help generating a test input file or setting up the word count example from your PPT as well.

**You said:**

hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ hadoop jar avgtemp.jar WordCount.WCDriver /rgs/avtemp.txt /out7 Exception in thread "main" java.lang.ClassNotFoundException: WordCount.WCDriver at java.base/java.net.URLClassLoader.findClass(URLClassLoader.java:476) at java.base/java.lang.ClassLoader.loadClass(ClassLoader.java:594) at java.base/java.lang.ClassLoader.loadClass(ClassLoader.java:527) at java.base/java.lang.Class.forName0(Native Method) at java.base/java.lang.Class.forName(Class.java:398) at org.apache.hadoop.util.RunJar.run(RunJar.java:321) at org.apache.hadoop.util.RunJar.main(RunJar.java:241)

**ChatGPT said:**

The error you're seeing:

cpp

CopyEdit

Exception in thread "main" java.lang.ClassNotFoundException: WordCount.WCDriver

means that the class WordCount.WCDriver is **not present inside your**avgtemp.jar**file**.

**✅ Here's How to Fix It:**

**🔸 1. Confirm the JAR matches the program**

If you compiled and packaged the **average temperature** program, then you should be using:

bash

CopyEdit

hadoop jar avgtemp.jar temp.AverageDriver /rgs/avtemp.txt /out7

**NOT** WordCount.WCDriver — that’s for the **word count** program.

**🔸 2. If you want to run WordCount**

You’ll need to:

* Save the WordCount Java files (WCMapper.java, WCReducer.java, WCDriver.java) in a WordCount directory.
* Compile and package them into a separate JAR file (wordcount.jar), like this:

bash

CopyEdit

# Save files into /home/hduser/Desktop/WordCount/ javac -classpath `hadoop classpath` -d wc\_classes WordCount/WCMapper.java WordCount/WCReducer.java WordCount/WCDriver.java jar -cvf wordcount.jar -C wc\_classes/ .

Then run the job:

bash

CopyEdit

hadoop jar wordcount.jar WordCount.WCDriver /rgs/avtemp.txt /out\_wc

**✅ Summary**

| **You Want to Run** | **Use This Command** |
| --- | --- |
| **Average Temperature** | hadoop jar avgtemp.jar temp.AverageDriver /rgs/avtemp.txt /out7 |
| **Word Count** | hadoop jar wordcount.jar WordCount.WCDriver /rgs/avtemp.txt /out\_wc |

Would you like me to generate the WordCount .java files for you so you can compile them and run it?

Top of Form

Tools

Bottom of Form

ChatGPT can make mistakes. Check important info. See Cookie Preferences.