

## Hadoop IP Log Count Assignment – Full Guide with Code and Commands

---

### Assignment Objective

Implement an IP count program using the Hadoop MapReduce model to count how many times each IP appears in a log file.

---

### Step-by-Step Guide

#### Step 1: Create Working Directory (Optional)

```
mkdir ~/iplog
```

```
cd ~/iplog
```

---

#### Step 2: Create Java Files

Create the following three files:

##### 1. UserLogMapper.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.mapred.MapReduceBase;  
  
import org.apache.hadoop.mapred.Mapper;  
  
import org.apache.hadoop.mapred.OutputCollector;  
  
import org.apache.hadoop.mapred.Reporter;
```

```
public class UserLogMapper extends MapReduceBase implements Mapper<LongWritable, Text,  
Text, IntWritable> {
```

```
    private final static IntWritable one = new IntWritable(1);
```

```
public void map(LongWritable key, Text value, OutputCollector<Text, IntWritable> output,
Reporter reporter) throws IOException {

    String valueString = value.toString();
    String[] fields = valueString.split(" ");
    if (fields.length > 0) {
        String ip = fields[0];
        output.collect(new Text(ip), one);
    }
}
```

## 2. UserLogReducer.java

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

```
public class UserLogReducer extends MapReduceBase implements Reducer<Text, IntWritable,
Text, IntWritable> {

    public void reduce(Text key, Iterator<IntWritable> values, OutputCollector<Text, IntWritable>
output, Reporter reporter) throws IOException {

        int count = 0;
        while (values.hasNext()) {
            count += values.next().get();
        }
        output.collect(key, new IntWritable(count));
    }
}
```

```
    }  
}  
}
```

### 3. UserLogDriver.java

```
import java.io.IOException;  
  
import org.apache.hadoop.conf.Configured;  
  
import org.apache.hadoop.fs.Path;  
  
import org.apache.hadoop.io.IntWritable;  
  
import org.apache.hadoop.io.Text;  
  
import org.apache.hadoop.mapred.*;  
  
import org.apache.hadoop.util.*;  
  
  
public class UserLogDriver extends Configured implements Tool {  
  
    public int run(String[] args) throws IOException {  
  
        if (args.length < 2) {  
  
            System.out.println("Please provide input and output paths");  
  
            return -1;  
        }  
  
  
        JobConf conf = new JobConf(UserLogDriver.class);  
  
        conf.setJobName("IP Log Count");  
  
  
        FileInputFormat.setInputPaths(conf, new Path(args[0]));  
        FileOutputFormat.setOutputPath(conf, new Path(args[1]));  
  
  
        conf.setMapperClass(UserLogMapper.class);  
        conf.setReducerClass(UserLogReducer.class);  
    }  
}
```

```
conf.setOutputKeyClass(Text.class);
conf.setOutputValueClass(IntWritable.class);

JobClient.runJob(conf);
return 0;
}

public static void main(String[] args) throws Exception {
    int exitCode = ToolRunner.run(new UserLogDriver(), args);
    System.out.println(exitCode);
}
}
```

---

### Step 3: Compile Java Files

```
javac -classpath `hadoop classpath` -d . UserLog*.java
```

---

### Step 4: Create a JAR File

```
jar cf iplog.jar *.class
```

---

### Step 5: Prepare Input Data

Create and upload a sample log file to HDFS:

```
echo -e "10.0.0.1 login\n10.0.0.2 login\n10.0.0.1 logout" > access_log.txt
```

```
hadoop fs -mkdir -p /iplog/input
```

```
hadoop fs -put access_log.txt /iplog/input/
```

## IF CSV FILE

### Step 1: Save CSV

Save it locally as network\_log.csv (with commas or tabs). If you're doing it from terminal, here's a way to simulate with a few lines:

bash

Copy code

```
echo -e "mac,ip,?,duration,?,start_time,mac2,end_time\n00-01-6C-D0-9F-  
25,10.10.10.221,1,8,1,\"01-06-2018 7.13\",00-01-6C-D0-9F-25,\"01-06-2018 7.40\""" >  
network_log.csv
```

Add the rest similarly, or copy-paste in a spreadsheet and export as CSV.

---

### Step 2: Upload to HDFS

bash

Copy code

```
hadoop fs -mkdir -p /network_logs/input  
hadoop fs -put network_log.csv /network_logs/input/
```

---

### Step 6: Run the MapReduce Job

```
hadoop jar iplog.jar UserLogDriver /iplog/input /iplog/output
```

---

### Step 7: View the Output

```
hadoop fs -cat /iplog/output/part-00000
```

Expected Output:

10.0.0.1	2
10.0.0.2	1

---

 **Notes**

- To rerun the job, delete the output directory:

```
hadoop fs -rm -r /iplog/output
```

---

**End of IP Log Assignment - Code + Commands**